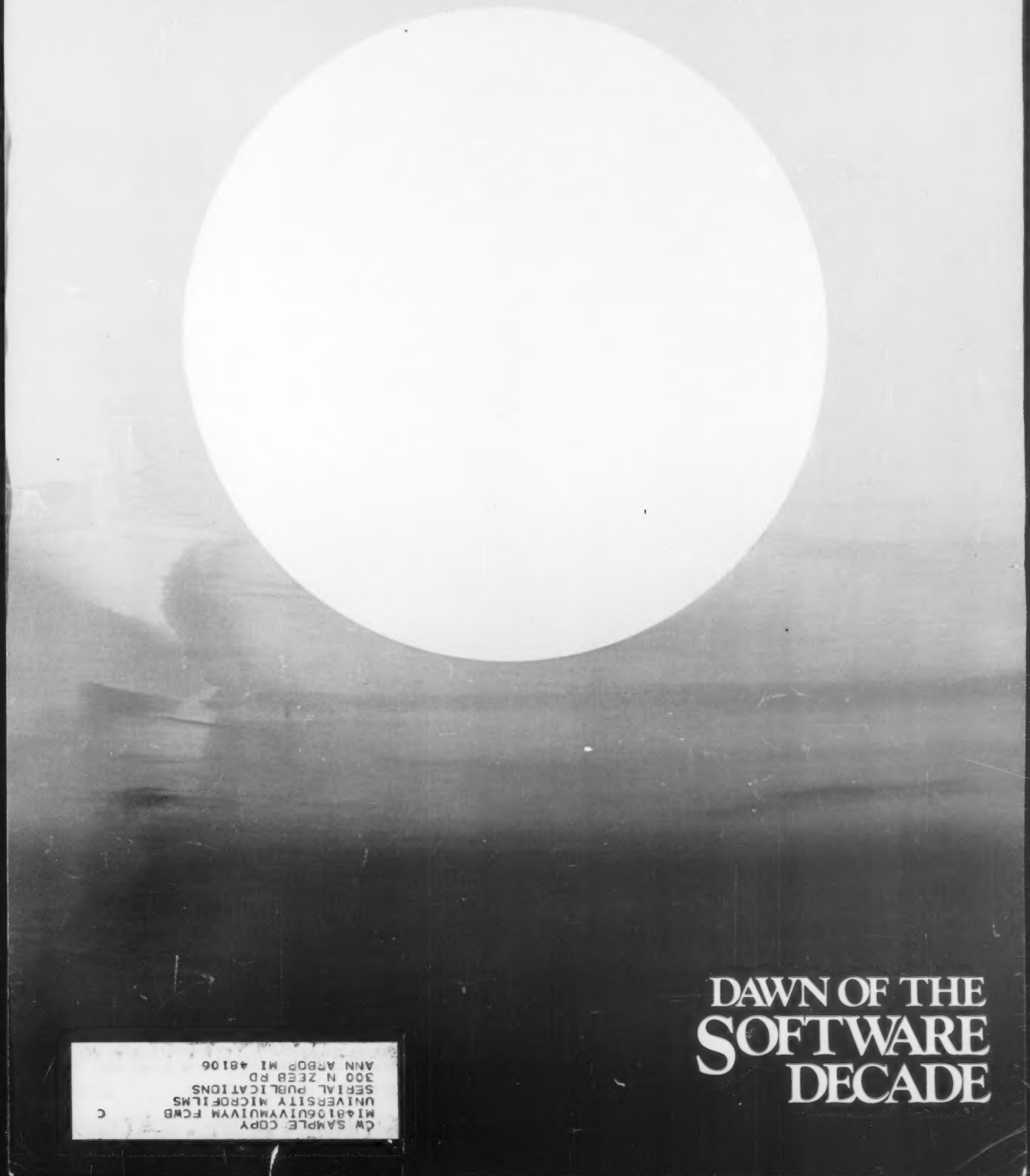
Computerworld Extra!

Vol. XIV, No. 38/Sept. 17, 1980

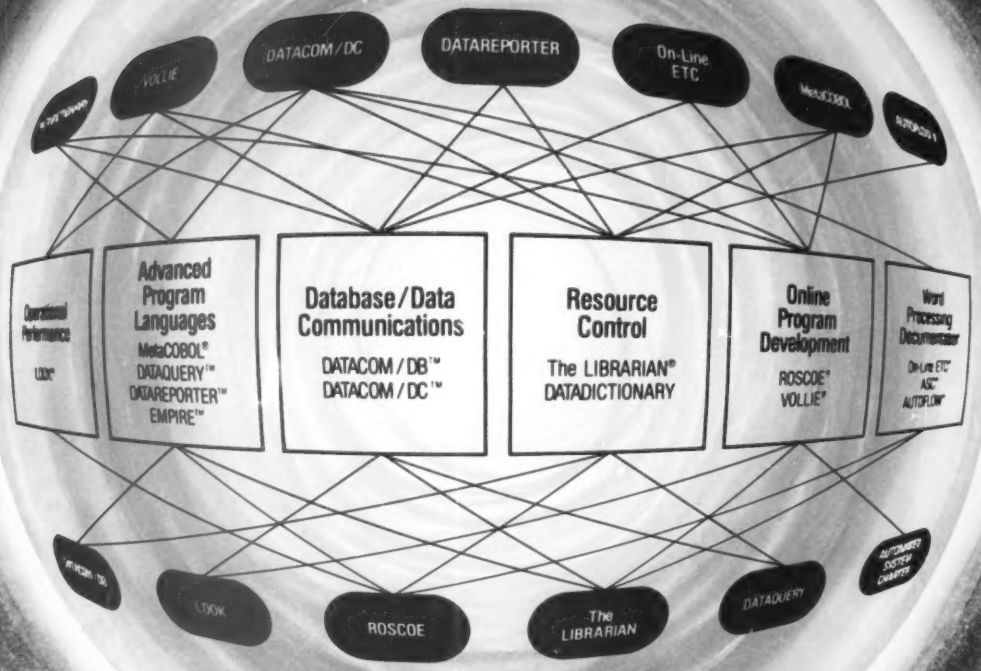
©1980 by CW Communications, Inc.



DAWN OF THE SOFTWARE DECADE

CW SAMPLE COPY
MI48106UIVYMUIVYM FCWB
UNIVERSITY MICROFILMS
SERIAL PUBLICATIONS
300 N ZEEB RD
ANN ARBOR MI 48106

From ADR...a new world of Integrated Software...



ADR's software product integration network offers a complete spectrum of state-of-the-art software for the 80's. By selecting ADR's integrated software products in the most meaningful way for their own environments, ADR customers will greatly reduce their entire program development and maintenance effort—without additional effort or overhead.

With over 11,000 product installations worldwide, ADR offers the most innovative, most integrated online systems software available from any source. This new dimension for the 1980's is your assurance that ADR software can not only help you solve today's problems, but enable you to build a solid foundation for addressing the problems of tomorrow.

For a new world of productivity, look into ADR's new world of integrated systems software.

at your fingertips

I am interested in:
- ☐ Call from salesperson
- ☐ Management presentation
- ☐ Brochure on ADR's integrated network

Name/Title _____
Company _____
Address _____
City/State/Zip _____
Telephone _____
Computer _____

APPLIED DATA RESEARCH

The Big Difference is our integrated network

Route 206 & Orchard Rd., CN-8
Princeton, NJ 08540
Telephone: (201) 874-9100

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 709 FRAMINGHAM, MA 01701

postage will be paid by

CIRCULATION DEPARTMENT

COMPUTERWORLD

375 Cochituate Road, Rte. 30
Framingham, MA 01701

Detach here, moisten and seal envelope securely before mailing.

Fold and insert order form (attached through binding) and remittance here.

CONTENTS

ON THE COVER: An original silkscreen by California artist E.T. Gipson captures the vibrant spirit of a new decade in software. With this special, bonus issue, *Computerworld* hopes to inform its readers of the many trends and developments that will shape the coming decade, which will truly be the era of software in the computer field.

Computerworld Extra!

Vice President/Editorial — E. Drake Lundell, Jr.
Extra! Editor — John C. Whitmarsh
Managing Editor — Nancy French
Art Director — Tom Monahan
Design Consultant — Garrow Throop
Typesetting Manager — Tom Pfau
Chief Copy Editor — Cheryl Gelb
Production Director — Donovan White

CW COMMUNICATIONS, INC.
375 Cochituate Road
Framingham, Mass. 01701

Board Chairman/Publisher — Patrick J. McGovern
President — W. Walter Boyd
VP-Research — Jack Edmonston
VP-Marketing — Roy Einreinhofer
VP-Finance — William P. Murphy
VP-Circulation — Margaret Phelan
VP-Production — Lee Vidmer

COMPUTERWORLD
THE NEWSWEEKLY FOR THE COMPUTER COMMUNITY

Second-class postage paid at Framingham, Mass., and additional mailing offices PN127420. *Computerworld* (ISSN-127-420) is published weekly, except: semiweekly the third week in September and a single combined issue the last week in December and the first week in January by CW Communications, Inc., 375 Cochituate Road, Framingham, Mass. 01701.

Copyright 1980 by CW Communications/Inc. All rights reserved. Reproduction of material appearing in *Computerworld* and *Computerworld Extra!* is forbidden without written permission. Send all requests to Marion Kibbee.

$1.00 a copy: U.S. — $30 a year; Canada, Central & So. America — $50 a year; Europe and Near East — $90 a year; other foreign — $150 a year (airmail service). Four weeks notice is required for change of address. Please allow six weeks for new subscription service to begin.

Computerworld can be purchased on 35 mm microform through University Microfilm Int., Periodical Entry Dept., 300 Zeeb Rd., Ann Arbor, Mich. 48106. Phone: (313) 761-4700.

POSTMASTER: Send Form 3579 (Change of Address) to Computerworld Circulation Dept., 375 Cochituate Road, Framingham, MA 01701.

5
Software in the 80s: Perils and Promises
By Daniel D. McCracken

Software development is, and will remain, the albatross around the neck of DP. But the limit is being pushed back, and quantum leaps are possible right now. The key: choosing the right tools. For years the whipping boy, the DP manager at last has a chance to become the hero of the organization.

11
Data Base: The Next Five Years and Beyond
By Stephen L. Robinson

There will be more effective use of data base systems, but for most organizations it will be the late 1980s before data base truly demonstrates its potential.

16
INTERVIEW: At Home with James Martin

James Martin is the foremost authority on the social and commercial impact of computers. In a wide-ranging interview, he tells Computerworld of the chief forces and developments shaping society and DP. He also gives us a glimpse of his life at home in Bermuda.

30
Applications Software Explosion
By Herbert L. Gepner

The use of applications software packages lagged in the 1960s, but has gained steady momentum since. Now the demand for every kind of off-the-shelf package is exploding. For the buyer overwhelmed by the range of products, here are pointers for selecting and purchasing packages plus a list of 100 important suppliers.

36
Data Management for the 80s
By Ronald G. Ross

The role of data dictionaries is expanding. Their encroachment into system design is gradual — but inevitable. The next step is assistance in implementing those designs, and that would represent the most significant change in implementation of computerized systems since the acceptance of Cobol.

48
Solving the Mystery of Data Base Design
By Thomas F. Meurer

Designers awash in the sea of data base know there are few strict rules in the logical and physical design of the system. A successfully structured methodology establishes a baseline design that is independent of hardware and software and that forces the production of documentation every step of the way.

52
Structured Methodology: What Have We Learned?
By Chris Gane & Trish Sarson

Structured methodologies have begun to crystallize only in the last three years. While project experience is still limited, some successful users are reporting significant savings in time and money. The secret, they find, is the construction of a graphical, logical model of the required system plus a statement of its objectives and constraints.

59
Open System Architecture and the Computerized Corporation
By Dixon R. Doll

Applications design should link users and data. Missing are pathways between the two.

65
Overview of Systems and Utility Software Packages
By William B. Engle

Don't expect dramatic breakthroughs in systems and utility software, but do watch for evolutionary enhancement. Here's an overview of the industry, including product assessments and two listings of major package vendors.

85
The Dimensions of Distributed Processing
By Burt H. Liebowitz

The proliferation of minicomputers has rekindled the debate over distributed versus centralized computing.

109
The Plight of Programming
By G. T. Orwick

The trickle is becoming a flood as the migration of programmers into application software companies leaves fewer and more expensive resources for DP managers. Despite new software development tools, an overall increase in programmer productivity is not expected to offset the loss.

Most installations have a 2-3 year backlog of unimplemented applications

Turns backlogs into applications

RAMIS II is a complete information management system. It integrates an English-like nonprocedural language with a flexible DBMS. The DBMS permits the easy integration of data from a variety of sources, while the nonprocedural language lets you tell the computer what you want done without having to say how to do it.

Users report that RAMIS II systems can be implemented in 1/5 the time it takes using procedural languages such as Cobol or PL/1. This translates into a productivity gain of 400%!

Because RAMIS II systems are simple to set up and modify, there is no need to develop elaborate specifications in advance. In fact, users report that basic requirements can be agreed upon and a prototype implemented in 1/3 the time it normally takes just to develop the specs. The data structures and reports may then be modified and enhanced in an evolutionary manner until the system is fully operational.

Eliminating programming also eliminates the need to talk about programming. This frees the user and dp staff to concentrate on the problem rather than the code–which results in both better communications and a better system.

In business, to stand still is to fall behind. More cost effective hardware, an increasing demand for computerized applications, and the decreasing availability of applications programmers means even bigger backlogs unless more powerful, more productive software is used.

For more information call or write today for a free RAMIS II factbook.

Name_____

Title_____

Company_____

Address_____

City_____ Zip_____

State_____

Phone_____

Mail to: Mathematica Products Group
P.O. Box 2392, Princeton, NJ 08540

RAMIS II SOFTPOWER

SOFTWARE THAT UNLOCKS THE POWER OF YOUR HARDWARE

MATHEMATICA 14 Washington Road • Princeton, NJ 08540 • (609) 799-2600

JARS USER NO. 2001

"I chose Johnson Systems' Job Accounting Report System for the same reasons 2,000 other DP Managers did."

Geoffrey Gumpert,
Tech Support Manager,
Beneficial National Life Insurance Company

"JARS gives me effective control over my costs."
Over 2,000 leading data centers rely on Johnson Systems' software to help them control cost accounting, computer utilization, workload management and hardware maintenance. And JARS is by far the most widely used system of its kind in the world.

"JSI maintains support for all OS and DOS releases."
From POWER/VS, IMS/VS, TSO and VM to CICS/VS; Johnson Systems is the only software vendor who maintains support for every OS-DOS release and option. That means control for today, and tomorrow, too.

"JARS takes only about an hour to install."
Getting up and running with JARS is surprisingly simple. Comprehensive working reports give you daily cost control information that lets you account accurately and bill equitably for all resources and users.

"JSI is the leader. And I always go with a winner."
For reliability and proven performance in data center management tools, nobody else even comes close to Johnson Systems' track record. So why bet on a dark horse? JSI is number 1 . . . the standard by which all other packages are measured.

"This time JARS even came with a bonus; five years free maintenance!"
Because they were JARS user #2,001, Beneficial National Life Insurance Company received, as a bonus to commemorate the event, five years free maintenance. Presenting the award certificate to Lou D'Andrilli, Vice President, Management Information Services at Beneficial National Life, are JSI President, Steven J. Johnson, and JSI Board Chairman, Paul Sterbutzel.

Johnson Systems, inc.

Mark Stoll · Johnson Systems, Inc. · 7923 Jones Branch Drive · McLean, VA 22102 · USA · Phone: 703 821-1700 · Telex: 89-9100
David Asdorian · Johnson Systems (UK) · 10 Station Road · Watford WD1 1EG, Hertfordshire · England · Phone: Watford 46255 · Telex: 851-923417
Ben Feitelson · Johnson Systems GmbH · Hunsrueckenstrasse 22 · 4000 Duesseldorf 1 · West Germany · Phone: 211/32 09 95 · Telex: 841-858-2177

CUT DOWNTIME · CONTROL UPTIME · FINISH ON TIME

- ☐ **OS JARS—Job Accounting Report System.** Performance measurement and computer accounting/costs distribution for OS users. (MVS, VS1, SVS, MVT and MFT)
- ☐ **IMS/VS—Utilization Monitor and Chargeback System.** An interface to OS JARS users, this is a powerful data reporting system for resource accounting and billing.
- ☐ **DOS JARS—Job Accounting Report System.** Accurate statistics on job scheduling, thruput, budgeting, resource utilization and software performance for DOS users. (DOS/VSE, DOS/VS, EDOS, DOS)
- ☐ **APEX—Automated Planning and Execution Control System.** Solves workload management problems by automating production control of OS data centers. (MVS and VS1)
- ☐ **UMAX—CICS/VS Utilization Monitor and Chargeback System.** Account accurately and bill equitably for all resources used by CICS/VS applications.
- ☐ **GOLD—CICS/VS Graphic Online Display System.** A UMAX option, GOLD permits the DP Manager to view component utilization, fine-tune performance elements, and monitor changes interactively.
- ☐ **ALARM—Hardware Reliability Measurement System.** Permits DP Manager to measure and reduce hardware/media failures by controlling in-house and vendor maintenance.

Please send me more information about the Johnson Systems Performance Products I have checked.

Name _____

Title _____ Phone _____

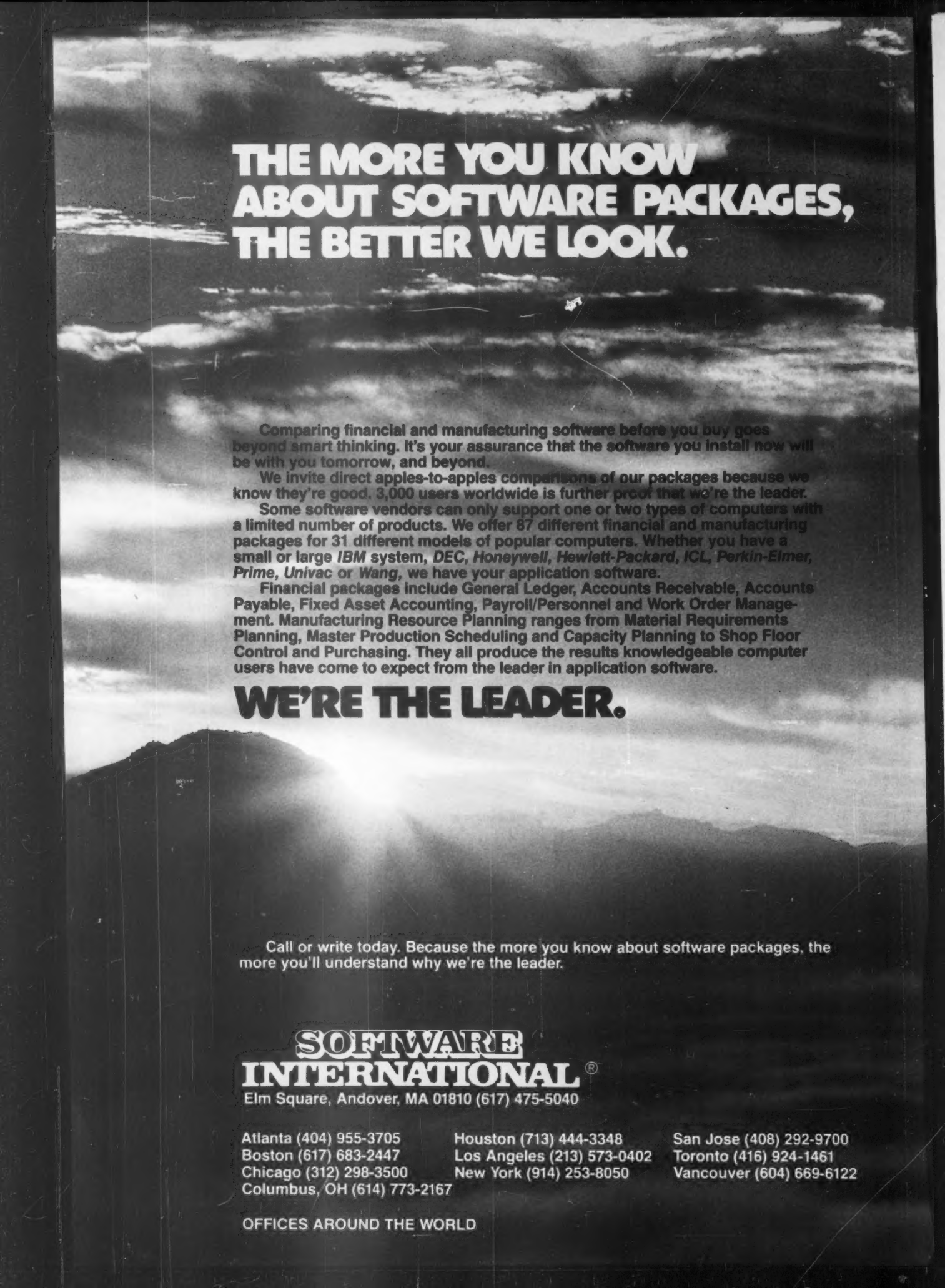
Company _____

Address _____

City _____ State _____ ZIP _____

CPU _____ Op. Sys. _____ TP Sys. _____

W010


THE MORE YOU KNOW ABOUT SOFTWARE PACKAGES, THE BETTER WE LOOK.
Comparing financial and manufacturing software before you buy goes beyond smart thinking. It's your assurance that the software you install now will be with you tomorrow, and beyond.
We invite direct apples-to-apples comparisons of our packages because we know they're good. 3,000 users worldwide is further proof that we're the leader.
Some software vendors can only support one or two types of computers with a limited number of products. We offer 87 different financial and manufacturing packages for 31 different models of popular computers. Whether you have a small or large IBM system, DEC, Honeywell, Hewlett-Packard, ICL, Perkin-Elmer, Prime, Univac or Wang, we have your application software.
Financial packages include General Ledger, Accounts Receivable, Accounts Payable, Fixed Asset Accounting, Payroll/Personnel and Work Order Management. Manufacturing Resource Planning ranges from Material Requirements Planning, Master Production Scheduling and Capacity Planning to Shop Floor Control and Purchasing. They all produce the results knowledgeable computer users have come to expect from the leader in application software.
WE'RE THE LEADER.
Call or write today. Because the more you know about software packages, the more you'll understand why we're the leader.
SOFTWARE INTERNATIONAL®
Elm Square, Andover, MA 01810 (617) 475-5040
Atlanta (404) 955-3705
Boston (617) 683-2447
Chicago (312) 298-3500
Columbus, OH (614) 773-2167
Houston (713) 444-3348
Los Angeles (213) 573-0402
New York (914) 253-8050
San Jose (408) 292-9700
Toronto (416) 924-1461
Vancouver (604) 669-6122
OFFICES AROUND THE WORLD

By Daniel D. McCracken

SOFTWARE IN THE 80s
PERILS & PROMISES

Software development remains what it has always been and probably will be for at least several more decades: the limit on what we can do with computers. Software development has improved, to be sure; higher level languages, structured methodologies, data base techniques — these and many other developments are significant steps forward that deserve to be used more widely than they are.

Still, by themselves, they are not enough. With rapidly dropping hardware costs and past successes creating vast new demands for computer applications, we simply cannot keep up by using existing software development methodology.

This is bad news in the obvious sense, of course, but there are two kinds of good news. For one thing, information processing is a victim of its own success. We're in trouble because users want more and more of the kinds of things we've already done.

The second item of good news is that the software development limit is being rapidly pushed back. We do not have to settle for the bleak prospect of mere incremental improvements in methodologies that will clearly never let us catch up. Quantum leaps are possible right now.

With continued intensive development of the new approaches, the level and sophistication of computer applications in the '90s could well look like another world.

The Problem

Presumably no reader of this publication needs convincing that we in applications software development have a problem. Everyone is aware of such symptons as:

- Backlogs of new applications and major changes that measure in the years *and are getting longer.* As an industry, we can't even keep up — let alone catch up — with what our users want us to do.
- Projects months and years behind schedule, vastly over budget and put into production with known major flaws or perhaps canceled outright after man-decades of effort.
- Incredible shortages of programmers and analysts, as indicated by a glut of recruiting ads, turnover rates reported at 30% to 40% and bounties for programmers ("Get your buddy to work here and we'll pay you $1,000 — if he stays six months").
- A dreadful crunch on the colleges, which see their present and future faculty being hired away so fast that their ability to produce more computer science and information science management graduates is probably *decreasing.*

We definitely have a problem.

The explanation that first leaps to mind is that we have a shortage of people. And we *do,* of course. Even if everything I'm predicting actually comes to pass, we will still need everyone now in the field and lots more besides. Anyone who wants to work in DP need have no fear of unemployment.

But a shortage of qualified people is clearly not the entire explanation of our troubles.

The root cause, I believe, is that to a great degree we are using the wrong applications development tools:

- We are writing unstructured code that is likely to contain bugs, be slow to get into production and be unnecessarily difficult to maintain — when structured methodologies have been amply demonstrated to ameliorate all of these problems.
- We are using Cobol and PL/I instead of much higher level languages, that, by comparison, are nonprocedural.
- We are writing what amounts to the same piece of code over and over, instead of writing it once in reusable form, or buying an application package or using utilities wisely.
- We are doing our own file management, security and integrity checking (if we do it at all), our own selection logic and our own report-writing functions — when a good data base management system, which includes a high-level inquiry language and a report writer, will do all of that and more with much less effort.

An analogy I like to use is that if you are using a hammer to drive screws, it really isn't going to help much to put energy into finding the right size hammer. Or deciding how to hold it. Or training people to use it. You're using the wrong tool, and that's all there is to it.

The analogy is imperfect, of course. The screwdriver was invented at the same time as the slotted head, whereas in our case we are still developing our tools. Maybe a closer analogy would be that we are still using screws and screwdrivers when what we need is glue or a new bonding process or a one-piece mold that doesn't involve *any* kind of fasteners.

The Solution

As a preliminary, since it underlies everything else I want to say, let me introduce what I will call "the software skill amplification principle." The assumptions are that (a) we don't have enough software writers, and (b) our methods are getting better but too slowly to solve the problem.

The proper response, I suggest, is to use our scarce software writers in such a way that their efforts are amplified, i.e., their work is made available to many others rather than applied solely to one application.

This is not a very earth-shaking concept; after all, it underlies things as old as subroutine libraries and reusable code. But it is coming into play in a wide variety of relatively new ways that have profound ramifications, such as software tools that put the computing power directly in the hands of the end user in many cases and let the DP expert bypass the wasteful step of specification writing in many others.

One characteristic shared by most of the new software tools and approaches that I want to discuss is that they involve what for lack of a better term is frequently called a *non procedural* approach. The term is unhappy; it defines an approach in terms of what it is not, and "procedural" isn't precisely defined.

Furthermore, a lot depends on one's point of view. In some sense, "Get born, live, then die" is a procedure.

Nevertheless, a sufficiently large difference of degree becomes a difference of kind. If a dozen lines of "code" containing the three "commands" LOAD, CREATE and LIST is a "procedure" even though it would take 30 pages and three weeks to do the same thing in Cobol, so be it — but you'll have to admit that it's a different *level* of "procedure."

Similarly, when a few options inserted into a data base schema can provide the automatic data validation — on both loading and maintenance —

Five Commandments for Developers

1. Don't solve any problem that has already been solved. Example: Don't write date routines, general ledger applications or your own analysis of variance routines — those problems *have already been solved.* Use utilities, buy packages or otherwise get software that has the capabilities you need.

2. Don't solve any problem twice yourself. Example: Don't write a program and then document it — use applications development methodologies that are self-documenting, such as very high-level languages or well-structured Cobol.

Writing the documentation after writing the program means that (a) it probably won't get done at all; (b) it is extremely unlikely to be fully accurate even when first written; and (c) it will almost never be up to date.

3. Don't solve any problem that someone else can solve better and more efficiently. Example: Encourage end users to develop absolutely every application they possibly can. Naturally, you'll have to provide them appropriate applications development tools to make this possible (see text for discussion).

4. Don't "solve" any "problem" that results in a net loss of time and resources in completing the application. Example: Don't write formal specifications except in those few cases where the need for them can be fully established; even in those cases, do so only after you have a prototype running.

5. Summary of above: Don't work harder than you have to, don't do any useless work and don't do any work that has negative value.

Daniel D. McCracken is an author, consultant and immediate past president of the Association for Computing Machinery. He lives in Ossining, N.Y.

INTERNATIONAL DATA BASE SYSTEMS, INC.

Another page from the SEED catalog

International Data Base Systems, Inc. 2300 Walnut Street Philadelphia, Pennsylvania 19103 215-568-2424

International Data Base Systems, Inc. would like to introduce you to our company and our complete line of data base management system software.

Since our establishment in 1977, we have developed a reputation for providing powerful, affordable software that is well suited for a wide variety of applications and hardware environments.

Our customers include large and small corporations, research labs, and universities. All, we feel, would be happy to talk to you about our products' performance.

All of IDBS's software share several features. It is easy to learn and use, compatible with many different computer systems, technologically advanced, and competitively priced.

We would appreciate it if you would take a few minutes to read the information in this advertisement, and consider the software products it describes for your possible use. We are confident you will find them worthwhile and are proud to present to you, this, our latest catalog of SEED products for your perusal.

Our most important product is SEED, an advanced and comprehensive data base management system that is easy to use, flexible, and inexpensive. In fact, we believe that SEED is more advanced and comprehensive than the better-known, more expensive data base products with which many of you who are reading this advertisement are already familiar.

What you should find most noteworthy about SEED is the fact that it combines great data structuring power and incredible ease of use in one data base software system.

SEED supports both network and hierarchical data structures and allows data base access from either COBOL or FORTRAN. It utilizes a CODASYL-compatible approach to enable your users to easily and confidently use SEED to develop new applications. You will also be interested to learn that SEED has been written in a combination of FORTRAN (for transportability) and Assembler (for efficiency).

Other useful features of the SEED system:

HARVEST is our query language and report writer, and is being used by many satisfied customers, some even "novice" computer users, to easily retrieve information from a SEED data base. With SEED and HARVEST, we have endeavored to make it simple for managers to get the information they need, when they need it, and in the form they want it. We are very proud of HARVEST and sincerely believe that using HARVEST you will find it quite simple to retrieve information interactively. You have only to specify the items in the data base to be displayed and the conditions under which they are to be selected. HARVEST will do everything else. We stand behind that statement, and can tell you of many satisfied HARVEST customers.

REAP is our report writer, and is designed to produce simple or complex reports from a SEED data base without the need for special programming, a feature which we believe you will appreciate. The report definition language is a non-procedural language for specifying the desired report. REAP was written so that your users do not need to know the structure of the data base. Report definitions may be stored for repeated use against different portions of the data base. Features you should know about are: control breaks of up to 15 levels; definition of temporary variables; complete page formatting; functions for total, maximum, minimum, average, standard deviation, and variance; and directed output to a terminal or printer.

GARDEN is our on-line data manipulation language that will aid the data base administrator and programmer. It provides facilities for interactive update and retrieval. We suggest you use GARDEN to verify procedure correctness, check data base integrity, fix errors in the data base, or create a test data base. Especially significant features include: prompting, conditional tests, "HELP" facility, looping, graphing, stored procedures, and on-line access to the data directory.

SPROUT is a multi-purpose file/data base conversion tool that allows transactions to be processed *into* and *out of* a data base. With SPROUT, you will find it easy to convert existing files to a data base. Or, if a data base is already in place, SPROUT can be used to create files for transport to a non-data base environment. Our customers tell us that they have found that SPROUT speeds conversion of data and eliminates the requirement for an immediate switch-over to converted programs.

SEED can be used in many ways around your shop.

We have learned of many ways to use SEED. Our satisfied customers tell us SEED has helped them improve their corporate planning, process control, general ledger, bills-of-material, engineering drawing control, inventory control, investment portfolio management, genetic research, scientific data collection, and employee benefit plan management.

We would be interested in learning about your needs, and how SEED might be able to satisfy them.

DBMS FOR VAX!

SEED is the oldest and most established DBMS on VAX -- having been available in native mode since early last year. Our many satisfied users tell us that the fine performance and service associated with this product continue to reflect our commitment to excellence.

SEED will flourish anywhere.

Our many customers throughout the world use SEED in many operating environments, and help to prove that our data base management system is as flexible as we say it is. SEED presently runs on

Digital Equipment	
VAX 11/780	VMS, UNIX
PDP 11	RSX 11, IAS, UNIX
DEC 10/20	TOPS 10, TOPS 20
IBM	
360/370/303x/43xx	OS, VS, VM, CMS, TSO
CDC	
6000 Series, Cyber Series	NOS, KRONOS, SCOPE
MODCOMP	
CLASSIC, IV	MAX IV
Hewlett-Packard	
HP3000	MPE

There's a small SEED too, for microcomputer users.

We have tried to be responsive to all our customers, and are making available, through Microsoft (whom we believe to be the leading microcomputer software company), MICRO-SEED, which is a compatible subset of SEED. It runs on microcomputers such as the 8080, 8085, Z80, and will soon be available on the 8086 and Z8000, and other popular products, utilizing CP/M and Microsoft FORTRAN. For information about MICRO-SEED please contact Microsoft directly at 206-455-8080.

We want to spread SEED.

We have made a commitment to help others build their products around SEED, and have put together a discount schedule that will brighten the heart of any OEM or system house.

IDBS 2300 Walnut Street Philadelphia, Pennsylvania 19103

I would like to learn more about the SEED data base management system.

- ☐ Please call me to discuss my specific requirements.
- ☐ Please send additional information on SEED and its features.
- ☐ I have no interest at this moment, but please add my name to your mailing list.

Name _____ Title _____

Company _____

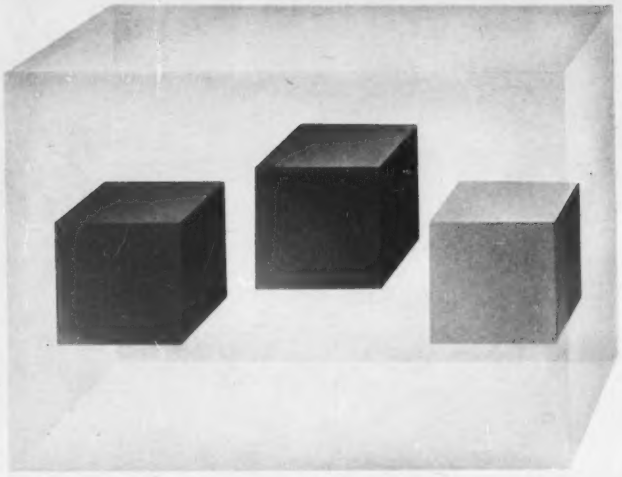
Address _____ Telephone _____

City _____ State _____ Zip _____

Computer _____ Operating System _____

SEED, HARVEST, REAP, GARDEN, and SPROUT are trademarks of International Data Base Systems, Inc.

The Transparent Single System Image Has Arrived!



With the recent addition of the Global Console Director (GCD) to Allen Services' line of software packages for Multiple System environments, an entire complex of systems can now be treated almost as though it were **one single unified** system.

The complete "unified system" effect is achieved through combined use of three independent (but related) program products. Each of these, Super-MSI, MSM and GCD addresses distinct concerns present in multi-system environments.

Of course, since the products are independent, if you aren't yet ready for the full-blown unified system effect, you only need to select the products which interest you.

UNIFIED DATA INTEGRITY (including VSAM)

SUPER-MSI™ The Multiple Systems Integrity Facility™

The Multiple Systems Integrity Facility provides the same thorough dataset integrity protection for jobs in **different** systems that would exist if the jobs were in the **same** system. In addition to propagating "standard" dataset integrity across systems, Super-MSI also insures that the VSAM "internal" SHR options are honored throughout your entire complex. Effective cross-system VSAM integrity can be obtained in no other way than through the use of Super-MSI.

Another feature: Super-MSI also **eliminates the hardware RESERVE lockouts** which are normally inherent with SHARED DASD in multiple system environments.

Much more besides: Super-MSI has many other exciting features including special aids for operators and TSO Users. All in all, Super-MSI will be a very powerful addition to your installation.

UNIFIED DEVICE MANAGEMENT

MSM™ MULTIPLE SYSTEMS MANAGER

The Multiple Systems Manager manages the allocation of TAPE and MOUNTABLE DISK devices across all systems in your complex. It allows you to operate normally with **all** devices **ONLINE** to **all** systems. MSM insures that device allocation by jobs in different systems occurs just as if all jobs were operating in one single unified system.

MSM eliminates the operational burden of juggling TAPE and DISK units among systems. It protects against the danger of a system accidently rewinding or overwriting a TAPE in use on another system; this removes the single greatest cause of shared tape mishaps.

MSM will use your TAPEs and DISKs more efficiently — which can result in immediate device cutbacks or (more likely) postponement of future acquisitions.

MSM is so transparent that it can be operated without the use of any new operational commands. Of course, there **are** powerful new commands available — however these are extras, not necessities.

Only MSM allows such truly NATURAL and transparent shared device management.

Other features (for MVS environments only):

- The MSM Device Preferencing capability allows devices to be dynamically structured into a hierarchy of sets; MSM insures that units in preferred sets are used, wherever possible, before less preferred sets.
- The MSM **SOFTSWAP™** feature insures that DDR device SWAPs for TAPE units will NOT impact JOBs or TSO users which are not using tape allocation. Without SOFTSWAP™, TAPE DDR SWAPS are likely to impede all allocations in a system — including even those which do not use tape.

UNIFIED OPERATOR CONSOLES

GCD™ Global Console Director

The Global Console Director is an extremely flexible tool. It allows (selected) message streams from different systems to be logically "blended" to create a **unified system image** (to whatever degree desired). In a slightly different vein, GCD allows **any** console on any system to be logically attached concurrently (and perhaps in different ways) to any set of systems in your complex!

GCD allows commands entered from any console to be directed and processed anywhere in the complex; in this respect, all consoles are effectively linked to all systems.

In almost every aspect, GCD is philosophically and operationally compatible with the standard system console support.

GCD can improve overall throughput by simplifying operations. By allowing several physical consoles to be logically merged, excess devices can be eliminated for immediate savings, or deployed elsewhere to strategic advantage.

GCD is absolutely transparent to your users, and aside from major simplifications, nearly transparent to operations.

If you wish to use more than one of the above products, they can be integrated together into a single control task (thus further reducing overhead and complexity).

In nearly every instance where our products have been competitively evaluated against all possible hardware or software alternatives — our software has been chosen. Aside from IBM, Allen Services has been providing commercial software specifically designed for multiple system environments longer than any other company. Over 400 major installations throughout the world have already selected our products as the preferable means of handling problems associated with multiple system environments. The number grows daily. **Reliability** has always been an integral part of our design.

When considering software, most people consider **support** to be a very important factor — Allen Services has over 200 support personnel.

The only other comprehensive approach to unifying multiple systems is JES3 (or ASP). Aside from the fact that MSI/MSM/GCD can be installed and implemented in only a few minutes, there are several other significant differences from JES3: For example, MSI/MSM/GCD (combined) use only 1% to 2% overhead; MSI/MSM/GCD are invisible to your users; MSI/MSM/GCD were designed to be strictly compatible with OS/VS/MVS philosophies — in this regard none of your personnel will require re-training.

The majority of all eligible multiple system installations are now using our products to help manage their systems.

Because installations can have widely different characteristics, you may still have a nagging suspicion that our claims may not actually apply to your shop. To alleviate this doubt, we invite you to try our products (whichever ones interest you) at your own installation under your own conditions. This trial carries no charge and is without obligation. One word of caution: You're going to like them so much, you'll never take them out.

ALLEN SERVICES CORPORATION®

For more information, call Susan: **800-543-7583** (in Ohio: 513-890-1200)
Or write: Software Department • Allen Services Corp.
212 W. National Rd. • Vandalia, Ohio, 45377

Abend-AID®

now over 750 USERS. WHY?

Abend-AID is the only sensible way to handle abends.

DIRECTORS

ABEND-AID will increase productivity and lower the cost in all areas of your data processing installation.
The product will produce hard dollar savings in excess of its cost in less than one year.

DATA BASE ADMINISTRATORS

ABEND-AID will simplify IMS development by providing complete diagnostic support for both IMS Batch and IMS On-line systems.

TECHNICAL SUPPORT

ABEND-AID will improve the performance of your CPU by eliminating the resources required to produce large memory dumps.
It will also reduce the time required of your staff in assisting with the resolution of application program abends.

OPERATIONS

ABEND-AID will produce immediate savings in the use of paper, ribbons, toner, etc.
The use of the product will also reduce printer time and increase your computer throughput.
ABEND-AID's English diagnostics greatly reduce the time spent on job recovery.

SYSTEMS and PROGRAMMING

ABEND-AID will shift the analysis burden from the programmer to the computer and will virtually eliminate the use of large memory dumps.
ABEND-AID contains comprehensive support for COBOL, PL1, ASSEMBLER, and FORTRAN.
ABEND-AID will reduce both your debugging time and the time required for test turnaround.

FREE SMF ANALYSIS

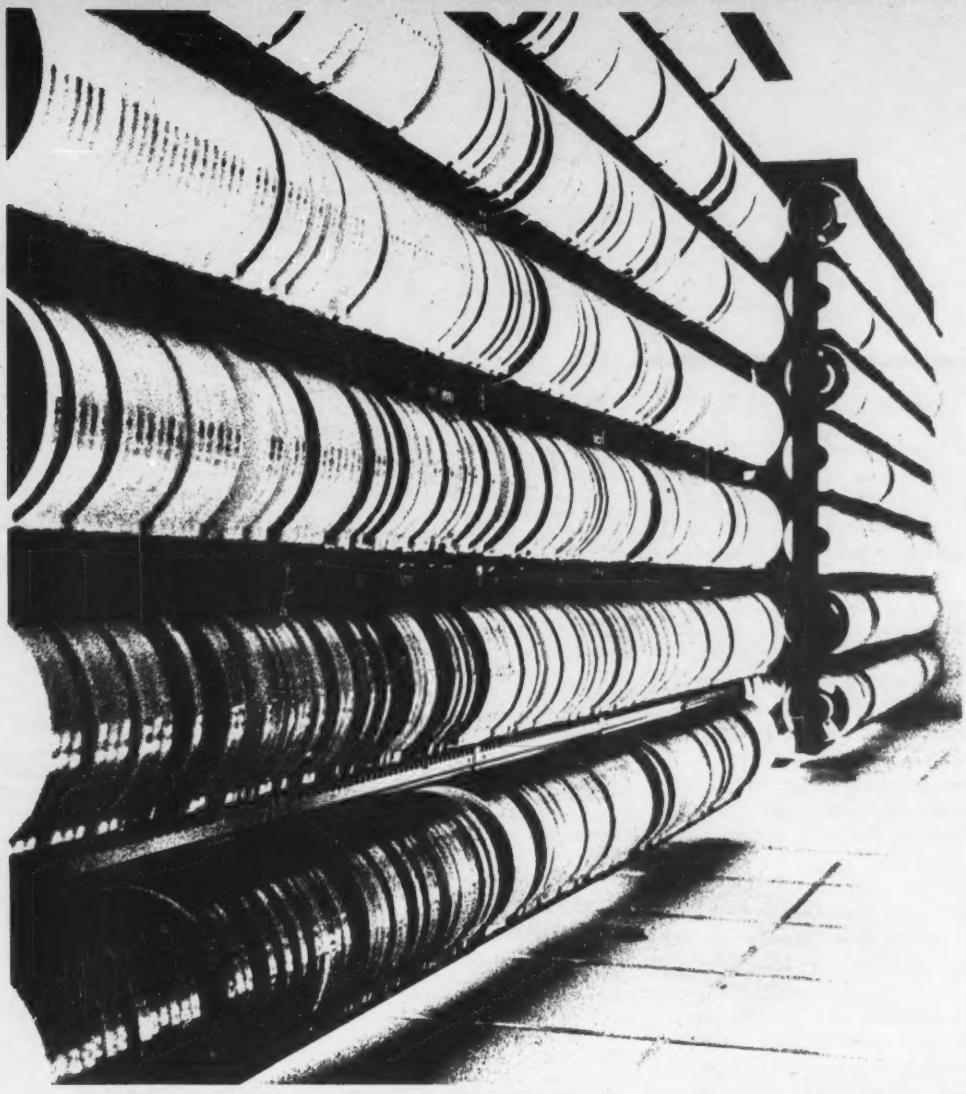
Call us at 800-521-9353 (In Canada call 313-559-9050).
We can provide you with our Free SMF Analysis Tape which will produce a report detailing the cost savings of using ABEND-AID in your particular organization. We also offer a 30 day trial of ABEND-AID at no charge.

TOMARK
29433 Southfield Road
Southfield, Michigan 48076
(313) 559-9050

ABENDABENDABEND
ABENDABENDABEN
ABENDABENDABE
ABENDABENDAB
ABENDABENDAB
ABENDABENDA

aid

Another Service of Compuware Corporation



that would take days of work to program laboriously in PL/I, we surely are dealing with a new kind of animal.

Let's see how these concepts come together to offer hope for a way out of the present applications development crisis.

An extremely promising approach to the shortage of applications software developers is to let the end users do the development. This approach also solves the underlying problem of communication between DP expert and end user, both of whom (necessarily) have their own specialized languages that require a translation. For applications on the smaller end of the spectrum, it is now possible to provide the users with sufficient power to let them specify the processing themselves, in systems that can be learned in a few days (really!).

I am most familiar with Nomad, a product developed under the direction of Harold F. Feinleib by my principal client, National CSS, Inc. (a Dun & Bradstreet Company), but there are a number of other systems that are to some degree similar.

These products are not identical (as their vendors will be happy to explain!), but all of them permit end users to do highly useful work in time scales that boggle the mind compared with traditional methods. Would you believe half a day to put up a focus benchmark version of a management reporting system for Heublein that had taken six months to do in Cobol? Would you believe that the Bank of America has something over 500 Nomad applications running, almost all put up by end users with the support of a staff of half-a-dozen Nomad consultants and training people?

I'm not quite sure what essential characteristics of a software tool let the end user do his or her own applications development.

Let me venture a short list anyway. The systems that permit the kind of results I'm talking about share these characteristics:

1. They are user-friendly, which, I take it, is a new buzzword signifying simplicity of learning, a natural syntax, helpful error messages, a default-driven design that makes simple jobs *really* simple and a nonprocedural approach that permits specifying a great deal of processing with a few commands.

2. They involve a data base management system that includes a high-level inquiry language, a sophisticated report writer and the ability to specify most of the data integrity and validation routines with nonprocedural options in the data base schema.

3. They are themselves large software packages, having taken away man-years, if not man-decades, to design and implement.

Naturally, not all applications development can be done by end users. There are always going to be big, complex applications with interdependent end users that demand the skills of the trained DP expert.

The traditional approach to such tasks usually involves, somewhere near the beginning of the process, the writing of a formal system specification. This can require man-months or man-years and result in a document at least a few inches — and not uncommonly *feet* — thick. The user is asked to sign off on it, and before he does so it is considered unwise — if not immoral — to begin program design or coding. An elaborate body of doctrine has been built up to prove that this is the only responsible way to do things.

The only trouble is, it doesn't work.

It doesn't work because users don't really know what they are agreeing to, for one thing. How is the user supposed to know that if he signs something saying "Dates will be stored in month-day-year order," that means we can't sort on the field? Why on earth should he be expected to know *that?*

Or how does the user know for sure that when he specified "A report of aged accounts receivable will be produced weekly," the systems analyst *really* understood what "aged" means?

The analyst *should*, so the user may not think to check his understanding — which can lead to major grief six months or a year later, when the misunderstanding has been cast into the concrete of an inflexible procedural program.

The communication problem here is the same one I mentioned in connection with end-user application development: both users and DP people have their own areas of expertise and their own vocabularies. When it is necessary for them to talk to each other, as it assuredly is on major applications,

Current Trends in Applications Development

1. Computing power is being placed directly in the hands of the end user. The process is limited mostly by human and organizational inertia. Applications development methodologies need to be tailored better for such use, of course, but fully operational existing methodologies that suit the purpose nicely are vastly underutilized.

2. Prototyping is coming into its own. Applications development specialists are learning how to get a version of even major applications up and running in a few days or a few weeks, and *then* they are worrying about performance.

3. "Nonprocedural" approaches are coming into much wider use. It just doesn't make sense to spend days writing a program in Cobol or PL/I or Pascal to do something that can be done with a few lines of commands in a much higher level language.

4. The "One Function in One Place" motto is being recognized as having wide applicability. It isn't *just* making parameters of the data of an application rather than scattering constants through the program (thus making changes extremely difficult and error-prone).

Much more broadly: Solve the documentation problem at the same time you write the program — by writing the program in a self-documenting methodology.

5. The attitude of data processing management is changing. More managers now are *welcoming* heavy end-user involvement, rather than feeling threatened by it. More have faced the real costs of getting work done and realized that the hardware costs are approaching negligibility compared with software development costs, the costs of long lead times and opportunity costs.

The Parable of the '67 Chevy

Once upon a time there was a man who wanted to be able to get from New York to Los Angeles on 12 hours' notice. His means of transportation was a 1967 Chevrolet that had bald tires and fired on three cylinders.

Recognizing the deficiencies of his means, he had the car tuned up and bought new tires, a supercharger and a radar detector. Performance improved 87%, but he still couldn't get from New York to L.A. in 12 hours.

He next put in a Cadillac engine and added streamlining; still not fast enough. His last gasp was to install a turbine engine and aerodynamic controls, which got him to 200 m.p.h. on the interstates — within reach of what he needed — but led to bad scenes going through the small towns.

When last heard of, he was complaining about problems with his turbine engine and small-town police; he had not realized that he was trying to improve the performance of a basically wrong means of transportation.

He was also asking what caused contrails and, when told about jet airplanes, said, "They'll never fly!"

Morals:

- **If you're using the wrong tool, improving its performance won't help much.**
- **The people who know the most about existing tools are often the hardest to get to accept better and different ones.**
- **The development of new tools seldom makes the existing ones totally obsolete. Even if our man had accepted jet airplanes, he would still have needed his car for getting to the grocery store.**

they have problems understanding each other and being sure they are talking about the same thing when they use a term.

Enter prototyping. Here is what *I* mean by *that* term:

Say you are a system designer. You get a call from a client who has a job to do on your computer, so you go to his office and talk for a while — a few hours, maybe a day. You try to establish clearly what the fundamentals of the tasks are, ignoring or almost ignoring data volumes, report formats, who authorizes changes and all that.

You go back to your office and put up a running prototype — which should not take more than a day or two ("should not" in the sense that otherwise the process isn't serving you or your client properly, and also in the sense that with proper tools there is no reason why it need take longer). You take your portable terminal to the client's office and say, "Gee, we talked an awful lot the other day, and I thought I'd just put up a simple version to see if I got it straight."

You sign on to the time-sharing system and let him try it. He says, "You know, that's amazing — it's almost what I want. Now, of course, you've got the columns backwards, and I forgot to tell you that I need subtotals, and the union rules changed since Tuesday, and anyway our conversation reminded me of some other things I'll need."

So you talk for another day, *with the running prototype to guarantee that you're both talking about the same thing*. Eventually you say, "Well, let me go back and play with this thing some more. I'll see you in a couple of days."

This iterative process continues until the client says, "You know, I just can't believe it, but that's the system I want! How fast does it run?"

OK, now you have a decision or two to make. If the whole job is going to run four minutes a month, you're done. Who *cares* if six man-months of programming would cut the running time in half? Doing any more work would be foolish.

On the other hand, if it's something that needs to run in no more than eight hours of every business day and your prototype version takes 20 hours, you need to do some more work. You have several options, depending on the circumstances:

1. You can go to someone expert in the implementation tool you're using and say, "Well, here's basically what I'm trying to do and here's how I did it. Can you make it run faster?" Maybe he'll see that you could have saved 30% by a different data base design and another 60% of the remainder by sorting the input. Maybe you're done. I know of instances where this step, taking a few days' time, improved performance by a factor of 20.
2. You can look at the implementation to see where the time-eaters are and reprogram them in PL/I or BAL or whatever it takes to get the necessary performance.
3. If the circumstances warrant, you can take the running prototype as the basis for writing a set of specs that is as complete as needed to support a complete reprogramming in some suitable language. (Ideally, software should produce the specs from the prototype. The only system I have seen or heard of with such capability is itself still in the prototype stage.)

In this latter case, where are you, compared with a more conventional approach? First you are very much more certain that the system you are about to design will actually solve the client's problem. *You've both seen something working*, and in the process of getting to that point, you've flushed out misunderstandings based on the language barrier. You've also flushed out more of the things he wouldn't have thought of until implementation, when they are harder to put in.

Second, the total time from start-up to finished specs will have been very much shorter.

Third, you have something that forms a far better basis for responding to changes — the inevitable changes — in the future. You can still use that prototype to try the changes out!

That's another thing fundamentally weak about procedural approaches and formal system specs: they resist change rather than responding gracefully to it.

Why isn't everybody doing it?

Fair question. Anybody who claims to have the solution to most of the world's problems — in any field of endeavor — should expect to be greeted by skepticism. I've made a few exaggerated claims for world-savers myself and had to eat crow later.

Well, the skepticism feeds on itself to a certain extent. Most people prefer to let some one else do the pioneering; once the passage of time has sorted out the wheat from the chaff, the good stuff can always be picked up later. Maybe that's smart. Surely not very many organizations can afford to try everything new, just to find the few goodies.

Another factor is inertia, which is perhaps another angle on the point just made. It simply takes a long time for a new idea to penetrate any field.

Another factor, if you'll pardon my saying so, is that it takes a long time for books supporting a new technique to appear. Something as basically sound as structured programming must have taken more than five years from inception to penetration of half the texts in the field.

Finally, in this listing of possible reasons for slow acceptance of these ideas (end-user applications development and prototyping), there is the opposition from the conventional wisdom. It isn't *just* that not too many people seem to know about these methods; there are still people going around saying that the methods are bad, wrong, unworkable, and maybe immoral.

Those of us who think as I do — and I am hardly alone and surely not the originator of the ideas — have our work cut out for us, spreading the good word.

Implementation Requirements

Three areas of change are needed to support extensive utilization of the newer approaches to applications software development that I have discussed:

1. More and better tool programs. This requires intensive research and development and will take time. For example, some systems provide good nonprocedural facilities for report writing and data validation, but force the designer back to procedural techniques for data maintenance. This particular problem is clearly solvable, but actually doing it takes work. In other areas further research is needed. This whole arena is a major illustration of the software skill amplification idea.
2. Wider acceptance by the technical people now in the field and more training support for end users and the students who will be coming into the field in the next few years. This, too, will take time. Resistance to change is to be expected and may in some ways be defensible. Those of us who claim to have seen the promised land always tend to be impatient.
3. A new breed of DP managers who *welcome* end-user application development, who are not threatened by it and who provide the facilities to do it (e.g., in-house time-sharing). I could name a few such now, if it were appropriate to do so here; there will need to be a great many more. And they, in turn, will need the support of top management — which will also need education.

Changes as fundamental as these will clearly require some years to implement. The scope of this special issue of *Computerworld* is the entire decade of the '80s, and I think that is probably about right; I expect that by 1990 today's conventional methods of applications software development will have been largely superseded.

Meanwhile, the new techniques are available for the using *today*.

The programmer or analyst who seizes the opportunity can be in on the ground floor of a major advance that is transforming our field.

And the manager who realizes what is within reach has an unparalleled opportunity to make a true leap forward in satisfying his users (finally!), in providing better service to his organization and in utilizing his scarcest resource — software development people — more effectively. After all these years of being the whipping boy, today he has the opportunity to be a hero in his organization. □

By Stephen L. Robinson

DATA BASE: THE NEXT FIVE YEARS AND BEYOND

For the last five years or so (and for the last three years in *Computerworld*), I have been making fearless forecasts concerning the next year's data base activities. I have had a significantly high rate of accurate predictions. There is a trick, of course, to such accuracy: a short prediction horizon.

When one concentrates in an area such as data base, it is relatively simple to forecast one year ahead, especially since most vendors usually announce new developments six months in advance — and leak them several months earlier. That leaves only a three-month span for intelligent guessing.

Now, however, I have been asked to write about the next 10 years in data base, not just the next year. Needless to say, looking 10 years ahead in an industry such as ours leaves one open to considerable error.

Accordingly, I have divided my predictions into three time intervals:

- Short term, an interval for which we expect the greatest accuracy.
- Intermediate term, for which accuracy is somewhat diminished, but still will hopefully be quite good.
- Long term, where a forecaster really succeeds or fails.

Actually, 10-year forecasts are good to make. If they prove accurate, you can resurrect them to demonstrate your success. If they prove inaccurate, no one will have remembered them anyway.

Short Term: 1980-1982

The future of data base systems will be significantly affected by future hardware developments. Our short term here is intended to cover a period of time in which no *radical* hardware architecture changes occur. This is not to say that hardware will not get more powerful and significantly cheaper; that will surely occur over the next three years.

However, I view this interval as one in which data base machines (see later discussion in "Intermediate Term: 1983-1985") will still not have made a major impact on the DP world.

I recognize that this interval may be too generous. Intel Corp. has recently announced the integration of its System 2000/80 data base software with its Fast-3805, a fixed-head disk memory replacement. In this configuration, all pointer tables for System 2000 will be placed on the 3805. This should markedly improve performance of the data base software. (At the time of this writing, Intel has published preliminary figures of a 50% to 100% improvement in performance, but I have not had an opportunity to experiment with the system yet.)

Given the definition of "short term," what can we expect from data base software? The biggest change by far should be an expansion of the role of data dictionaries. At present, many vendors of data base packages also market data dictionary packages. The packages are, for the most part, quite limited in scope.

The typical data dictionary package today is an "off-line" utility to provide services such as:

- Storing data about an organization's data and the attendant areas of interest (eg., reports, files, projects, people, etc.).
- Selectively querying the above for information about the data base environment (eg., what file contains the field product number; what programs update the field department, what fields are accessible via the password PAYR2815; etc.).
- Maintaining control cards to run various configurations of the data base package.
- Maintaining "canned" data base procedures, usually for report writers or query languages.

While the above are quite useful, it is clear that most of the data base package vendors are working toward the configuration shown in Figure 1. Here the data dictionary has been truly integrated into the data base package. Indeed, it might be said that the data dictionary is the controlling package in such a configuration.

The functions attributed to the data dictionary interface are, of course, only representative of the most likely areas that will be implemented in the time frame under discussion. Indeed, we have "cheated" a bit in that several of the functions have already been implemented in several integrated data base/data dictionary environments.

Query Language

It is clear that the data base package vendors are making major development efforts to enhance existing query facilities, with a view toward the evolution of true natural language query facilities.

In short term, the major benefit in this area should be for DP personnel, not end users. System development times should be reduced through effective use of query languages but, unfortunately, increased use of these facilities will also promote the idea of reduced system analysis time, which will certainly lead to problems in the second half of the '80s.

Ever since it was claimed that Cobol and Fortran would eliminate the need for programmers, end users have been regularly promised natural language query facilities. At present there are numerous packages which claim to be "user-friendly."

The packages which come closest to earning this label are the simplest retrieval systems. Basically, such systems provide minimal flexibility in record qualification and display and thereby require minimal grammar constraints.

By contrast, some packages combine extensive report-writing capability with true programming language facilities. These systems are not "user-friendly" and should only be utilized by DP personnel.

The latter class of package is very useful in computer personnel's generation of "canned" parameterized modules which can be stored in a library and called by non-DP personnel who have only to conform to minimal constraints on the entering of required parameters.

During the next several years, the major area of query language development — combination programming languages and report writers designed to optimize the use of data base facilities by DP personnel — should be discussed.

It appears unlikely that we will see significant strides in data retrieval languages for non-DP personnel.

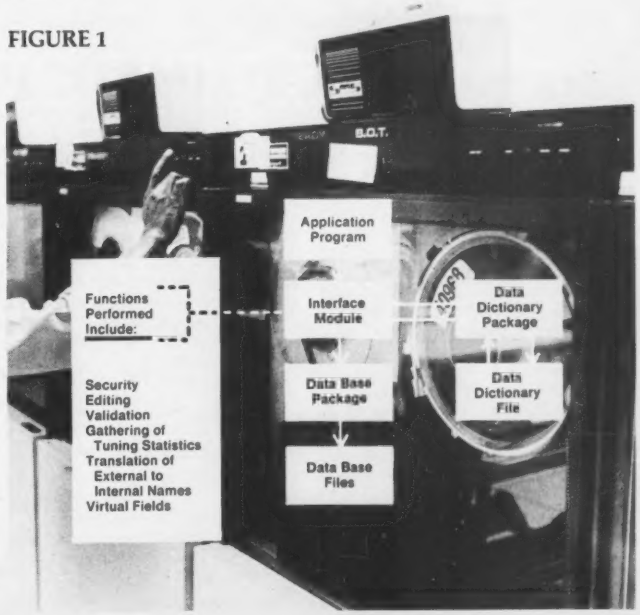
Possible events that might negate this view would be the rapid development of word processing interfaces; and a significant overlap of word processing and DP facilities and personnel; and a proliferation of major data bases implemented for minicomputers and microcomputers.

One last thing should be mentioned in conjunction with query language development and use. Such facilities are currently being used in much the same manner as data base packages. That is, application-oriented files are being processed in isolation from the remainder of an organization's data.

This is usually done in one of two ways. Either additional indexes are created for production files (thereby creating additional overhead for the maintenance of the production files), or a spin-off file is created from the production file, solely for query purposes (usually resulting in querying against "almost up-to-date" files not to mention the computer resources required to create the spin-off files).

If our estimate of three to five years for a significant shift from file-oriented systems to data base-oriented systems is correct, the same time frame should roughly predict a switch to interapplication use of query facilities. Almost unnoticed

FIGURE 1



Stephen L. Robinson is president of S.L. Robinson & Associates, a consulting firm in Yardley, Pa.

Robinson will present this paper at the Infotech State of the Art Review to be held Nov. 26-28 in London.

amidst the hoopla of data base systems, data banks are now truly coming into their own. Data banks range from the trivial (automated mailing lists, usually with at least minimal selection capability) to the complex (economic forecasts broken down by industry and time interval); from the small (several thousand entries) to the enormous (millions of entries).

Until recently the use of most such data banks has been limited by system design; that is, usage was batch-oriented. Clients would submit requests and get printouts in return. Browsing and heuristic analysis were not easily accomplished.

This is now rapidly changing. Systems are being implemented that encourage on-line utilization of data banks and associated data analysis software (e.g., statistical packages).

"Bare-bones" data retrieval systems are being replaced by extensive information systems. Where economic data used to be available on various industries, now information is available on the interaction of different segments of the economy, computed at varying interest rates, different international assumptions, etc.

The next several years should witness an expansion of information systems based on data banks. Financial systems will lead that development; marketing and communication/transportation systems should follow. Pandora's box will then open to release virtually every nonautomated data bank.

Intermediate Term: 1983-1985

I have already discussed the significance of the lower boundary for this period — the advent of important hardware architecture changes. Without predicting the nature of such changes (associative memory, high-speed mass storage and large-scale directly addressable memory, for example), I will address the processing environment that might result.

The hardware changes should markedly improve our ability to store, index and retrieve data. Given such changes, performance criteria will become progressively less important. During the early part of this period, we will see an embracing of the heuristic systems design methodologies now beginning to appear in our industry.

It will not take the industry long to realize that there are indeed no shortcuts to good system design. Rapid development and implementation of faulty systems working off of erroneous data will once more result in an angry user community.

By 1985 or 1986, it should become evident to corporate and organizational management that the major obstacles to the development of data base systems are organizational, not technical. This should result in the most significant change in data and information processing since the development of written symbols and numbers. The upper bound for this interval, therefore, is intended to reflect a timetable for the transition of our industry from solely data processing to data management.

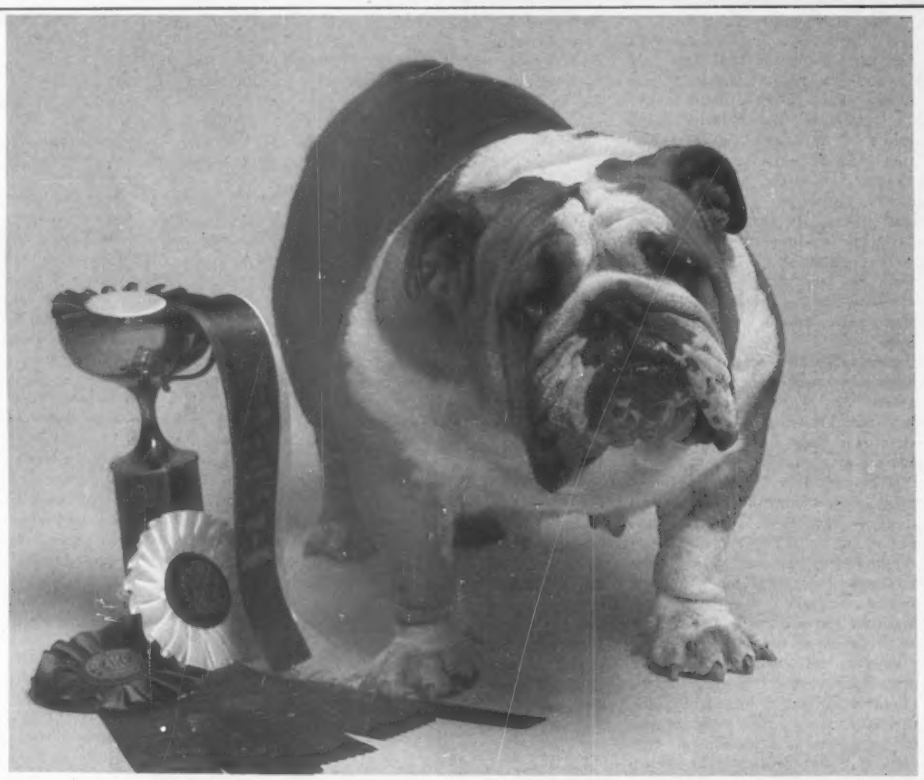
Organizational Changes

It has been 25 years or so since automated data processing made its first major inroads into the business world. Electronic accounting machine (EAM) equipment rapidly gave way to sophisticated second-generation computers. The DP industry was well under way.

DP as a profession has been somewhat slower to mature. We are still mired in the three-step operator-programmer-analyst career path. This despite all the evidence pointing to the fallacy of such procedures — better programmers tend to be introverted, better analysts extroverted; better analysts thrive on unstructured, ill-defined problems, better programmers work best on well-defined problems, and so on.

Of equal significance is the slow, almost nonexistent progress toward the recognition that there is more to DP than the creation of working programs and systems.

It is clear that there is a growing awareness that data, and resultant information, is a corporate (organizational) resource, and therefore should



Success Is A Matter Of Performance

In today's computer environment, systems are so complete . . . and so dynamic . . . that attempts to maintain in-house development of systems software are becoming counter-productive. The fundamental fact, that has been accepted by successful data processing center operations, is that not every center has the capacity to produce each and every software tool that the present expanding hardware utilization allows.

The "communication age of the '80's" establishes a *data processing society*, that calls for a formal, yet workable, mode of operation.

Hardware capability has given way to software performance as the criteria for successful data processing operation.

At Software Module Marketing, performance is *Where Service and Software Come Together* . . . More than a slogan, SMM believes *service* is a commitment to provide our customer with marketing and technical support that will continue to advance their leadership standards.

. . . When you have a product from SMM, you can count on three things — *Productivity, Service and Performance.*

SMM products support OS/VS1, MVS, MVS/SE, MVS/SP operating systems on IBM and compatible mainframes.

DMS/OS — The disk management system that optimizes the use of DASD resources.

ACEP — The Conversational On-Line Time Sharing System.

STAM/SDSI — Integrity systems supporting the multiple CPU shared tape/disk environment.

Software products serviced and marketed by SMM. Call, telex, or write today for a free thirty-day evaluation, or additional information.

CROCKER BANK BUILDING/PENTHOUSE
1007 SEVENTH STREET SACRAMENTO, CALIFORNIA 95814
(916) 441-7234 TELEX: 377314

SOFTWARE MODULE MARKETING

Where Service and Software Come Together

SCANDINAVIA • BENELUX • FRANCE • U.K. • GERMANY • ITALY • AUSTRALIA • JAPAN

"The trickiest problems to resolve will be those where functions overlap between the areas of MIS, DP and data base."

be managed. DP departments, as noted above, do not manage data, they process it.

Progress is slow. Recognition usually begins in the DP area. Given the funds to acquire data base, teleprocessing and related software and given the charter to create a data base system, most data base administrators quickly realize they do not have authority to implement interfunctional area decisions.

Without such authority they cannot develop the systems they have been promising to the user community. Intent on keeping their jobs, data base administrators rapidly fall into the file system trap. That is, although they recognize the long-term problems, they use a data base package as a sophisticated access method to reimplement existing systems.

There have already been several well publicized, costly "data base failures." Invariably such failures are not failures of data base, but failures to recognize its organizational implications.

Several years after management has paid for new hardware and software and still does not have the data information systems it requires, we should see the slow but certain transition to the type of high-level data administration function that presently exists mainly in the literature and rarely in practice.

Figure 2 (see Page 15) shows a possible positioning for data administration in an organization. Many such organizational charts can be hypothesized. Especially in the early days of this period, organizations will be experimenting with reporting lines.

The trickiest problems to resolve will be the responsibilities below the vice-president level, where functions overlap between the areas of MIS, DP and data base. To date, corporate management seems willing to let the areas "fight" over the overlap. Eventually, hard decisions will have to be made to ensure effective cooperation between the three groups.

These problems will not be resolved by 1986 in the majority of organizations. However, they will be recognized, and will therefore be on their way to resolution.

Software Changes

The literature today is filled with articles on relational data base structures. This proliferation of articles, more than anything else, has led to a great misunderstanding on the nature and state of development of relational data base systems.

First, some software vendors have managed to convey the impression that relational data base is a physical concept they have implemented and are presently offering in the form of a relational data base package.

Second, the preponderance of theoretical articles on relational concepts and the relative validity of third, fourth, fifth and sixth normal forms has led many people to regard relational structures as the province of academicians and theoreticians with beards, long hair and sandals.

In reality, of course, relational structures represent a means of logically representing data structures. Other logical modeling techniques exist (infological modeling, entity-attribute-relationship modeling, etc.). However, the support being given by IBM to Codd's work suggests that third normal form structures will dominate modeling efforts.

Direct physical implementations of relational structures are impractical with today's hardware for the representation of most organizations' data. However, as noted in the discussion of the year delimiters for this period (1983-1985), I anticipate that 1983 may well be the time horizon for significant hardware changes in particular, changes which may facilitate the implementation of relational structures.

The advent of relational packages will not, however, ensure the development of integrated data base systems. As noted in the discussion starting this time period, practical physical relational structures should precede the recognition and acceptance of true

DBMS
CDPS
HOW TO FLY
HOW TO CRASH
Other
SERIES 1
DATA BASES
are OPERATIONAL...
ARE YOU STILL WAITING?...
The WISE have it . . . CDPS — The Data Base/ Distributed Connection for IBM — Series/1.
• THE ONLY COMPLETE DATA BASE FOR IBM SERIES/1 . . . WITH ALL SOFTWARE (Utilities, System Software . . . Everything!) TO DO THE JOB . . . where it counts.
• CONTROL INFORMATION IN THE FIELD . . . WHERE IT IS NEEDED — Transport files to be used by programs at remote locations (standalone, offload, integrated network . . .)
• CONTROL THE PRODUCTIVITY OF YOUR DATA PROCESSING FUNCTION — USERS STATED CDPS reduced development time BY SEVEN TO TWELVE MONTHS . . . INCREASED PROGRAMMER EFFICIENCY BY 33% . . . REDUCED DATA TRANSMISSION COSTS . . .
OUR FREE DIAL-AN-ANSWER HOTLINE IS HOLDING FOR YOU . . . CALL IT FOR THE CONVENIENT CDPS SERIES/1 CONNECTION !
TOMINY INC.
4152 Crossgate Drive / Cincinnati, Ohio 45236
HOTLINE: 1-800-543-4624
(OHIO) 1-513-984-6605

"It is not at all clear that a true natural language facility is what is really needed."

data administration. Consequently, it is likely that relational data base packages will initially be used simply as access methods to underlie relational query languages.

Query Language/Data Banks

There are many people who feel that relational query languages, when practical, will revolutionize DP. The end user will finally be freed from the necessity of communicating with an analyst who, while proficient in the language of computers, doesn't really understand business procedures.

There are flaws in such expectations — perhaps enough flaws to completely negate the potential value of such query facilities.

First, most non-DP personnel, while quite capable of following simple grammar rules to define the characteristics of records of interest and the format of a report, are quite incapable of writing procedural statements for even simple business algorithms. This is not attributable to any shortcomings on their part, but rather to a distinct lack of natural language flair in the present programming languages, report writers and query languages. It is unlikely that these problems will be resolved in the next five to seven years.

Second, it is not at all clear that a true natural language facility is what is really needed. Marketing people, after all, speak "marketingese," manufacturing people speak "manufacturingese," and so on.

Take a simple word like "extend." It can take on half a dozen meanings depending on the application area of the person using the term.

The ideal query systems may very well turn out to be the ones in which simple data querying can be done by non-DP people with little or no ambiguity, and complex programming can be done by DP people to create parameterized programs for execution by non-DP people. At present, several such systems are in existence (eg., Software AG's Natural language), and we can anticipate further enhancements and refinements to make them more user-friendly.

Crystal Ball: 1990

It is 1990. You have just finished a project ahead of schedule. Your manager suggests you take a couple days off without reducing your annual leave.

You fight the evening rush hour traffic (some things won't change by 1990) and eagerly head for your home entertainment center (for such will be the name chosen by the marketing people — witness the already growing reluctance to call a microcomputer a computer). You activate ("log on" sounds too computerese) the vacation information channel from your Comm Station (sounds better than "keyboard"), thereby initiating charges to the appropriate service vendor.

Snow conditions are ideal in Davos and crisp, sunny days are forecast for the next week with a likelihood of snow for five of the evenings.

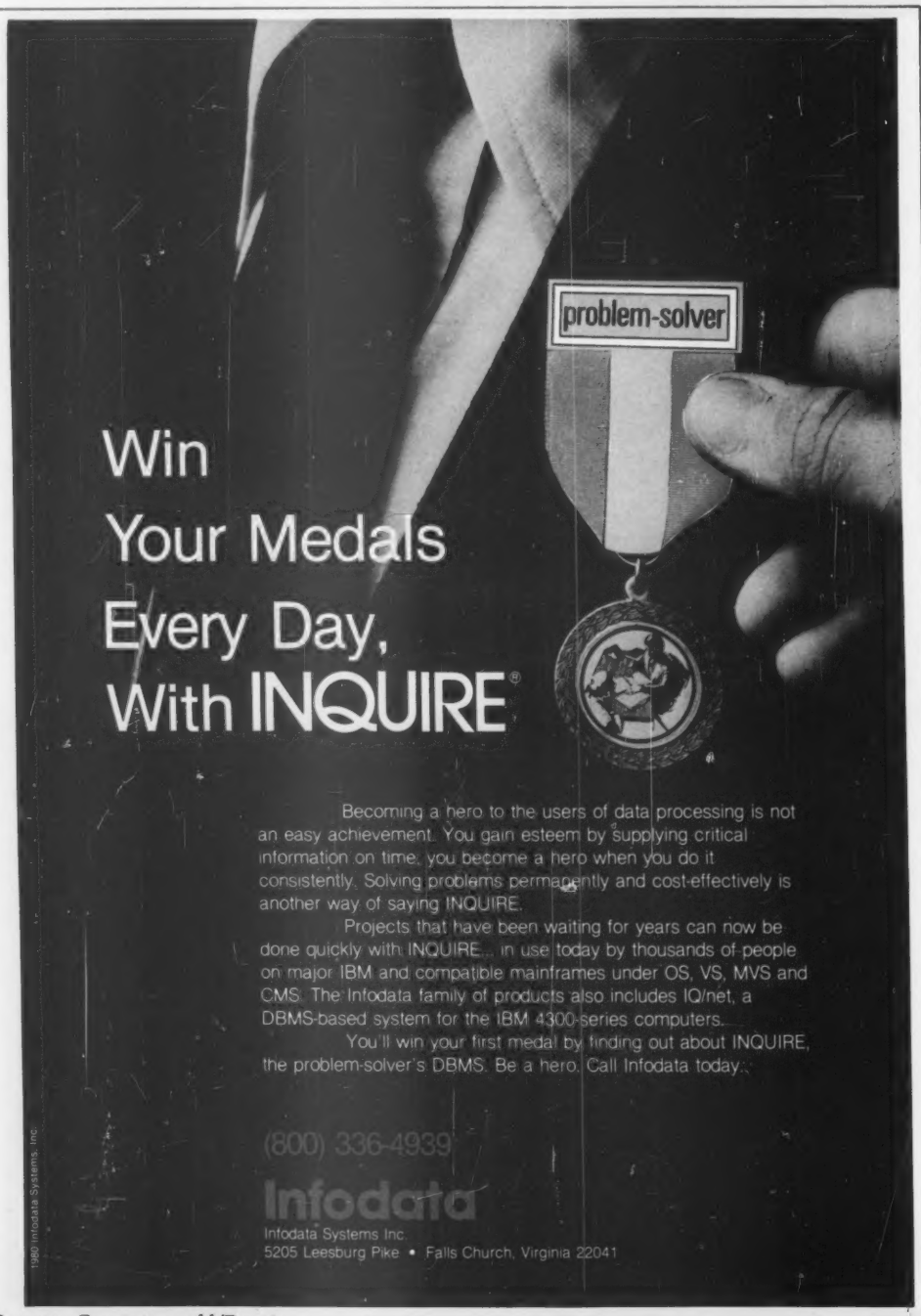
You cancel the vacation channel and dial a reservations channel. In rapid order you arrange for air transportation, ground transportation, lodging and ski lessons.

Sounds farfetched? It's not. The reservations facilities, of course, are here now. Ski conditions are available through telephone dial-ups. Add a microcomputer to tie together the Comm Station, the wall television and the telestation (or pick your own name for the telephone of the future) and you have the system just described.

A competent microcomputer hobbyist could build the necessary home hardware configuration for less than $1,000. The existing cable TV stations, if tied into satellite transmitters, would provide the requisite channel selection for the system.

Want to buy some ski clothes for the trip? New equipment? Transfer funds? All done from the convenience of your home. Again, farfetched? Not really. Likely? Perhaps not; like everything else, such facilities will have to be able to pay their own way. Their cost in of 1990 may not justify their implementation.

So much for the bright side of 1990.


problem-solver
Win
Your Medals
Every Day,
With INQUIRE®
Becoming a hero to the users of data processing is not an easy achievement. You gain esteem by supplying critical information on time; you become a hero when you do it consistently. Solving problems permanently and cost-effectively is another way of saying INQUIRE.
Projects that have been waiting for years can now be done quickly with INQUIRE... in use today by thousands of people on major IBM and compatible mainframes under OS, VS, MVS and CMS. The Infodata family of products also includes IQ/net, a DBMS-based system for the IBM 4300-series computers.
You'll win your first medal by finding out about INQUIRE, the problem-solver's DBMS. Be a hero. Call Infodata today.
(800) 336-4939
Infodata
Infodata Systems Inc.
5205 Leesburg Pike • Falls Church, Virginia 22041
1980 Infodata Systems, Inc.

Let us now turn to the grim reality of the business world. Jean Smith, director of international sales for the ABC Co., is considering the expansion of a product line. She consults her corporate data base system and discovers that:

- Inventories of the product increased by 175,000 units in the last quarter.
- Production was 1.2 million units last quarter.
- Sales were 1.4 million units last quarter (note the contradiction in the preceding three statistics).
- Entering PRINT SALES results in a thoroughly confusing error message "IRROE2745-Invalid Operator," or nonindexed key or anomolous word sequence.
- Kicking the base of the terminal in response to the above message, banging several random keys and hitting ENTER results in a strange message about a SOC4, from which it is necessary to telephone the computer center to recover.

In short, the business world may be little better off than it is now. Let us try to identify why this may be true in order to avoid such problems.

If we do not recognize that DP personnel are notoriously poor at human engineering, we will continue to provide the outside world with interfaces that look like JCL (possibly the worst travesty ever foisted on our industry). We must realize that human engineering is a discipline unto itself, and we must develop people capable of producing user-friendly systems.

We must begin to draw our system boundaries larger. It is unsatisfactory now, and will be even more so in the future, to have separate and independent order entry, manufacturing, general ledger and warehouse systems.

Data administration must be recognized as distinct from DP.

We should understand the different skills required from application programming, system programming, computer analysis, business analysis and computer project management. We should accordingly establish career paths which allow personnel to pursue their interests and talents without suffering financially.

We *must* stop using new hardware and software to emulate (simulate) older hardware and software systems. It's a little like an airline which has completely changed its fleet from props to jets, but maintains the old schedule.

EAM Systems

The future of data base systems is very much up for grabs. In the beginning DP personnel designed systems based on EAM equipment. When tape-based computers arrived on the scene, very little new systems analysis was done.

Instead, process boxes with "read a card" were replaced by boxes with "read a record." Records were invariably 80 characters long (or multiples of 80). Systems tended to be an endless succession of sort/print operations.

Along came third-generation computers. The wave of the future. But, alas, along with them came emulation. Why perform new systems analysis and design when old programs could be emulated? Disks could be made to look like tapes etc.

Along came data base software. Another opportunity for change. Another failure. Yes, failure. Data base software has been around for 10 years, give or take a year or so, yet accomplishments are few and far between. An estimated 75% to 90% of data base package installations are using the software as a sophisticated access method to reimplement file system logic which at the source was probably EAM logic.

Data base software and hardware will continue to evolve. Quite honestly, I don't believe they will be effectively utilized until late in the 1980s. We must overcome 20 years of DP inertia before the direction of our profession can change. It will be a struggle, a challenge; with a bit of determination, we will succeed. The alternative is continued chaos.

What's Missing

The discerning reader has probably noted several omissions in this article — for example, distributed data base, the latest industry buzzword. Where to place it is a bit of a problem. Software is already starting to appear which offers some capability which could be called distributed data base. Are we already there? Where is there?

Definitions of distributed data base abound. For the purposes of this article:

- Decentralized data processing — autonomous computer centers.
- Distributed data processing — physically disparate computer centers under *central* control. The separate centers may or may not be physically connected by channels or other communications lines.
- Decentralized data base — autonomous data bases usually representative of individual functional areas.
- Distributed data base — physically disparate data bases under *central* control.

The term "distributed data base" is unfortunately used today to refer to autonomous data bases under a single data base package or manager. "Unfortunately," is an appropriate interjection since such an environment inevitably leads right back to file systems concepts.

Also, without a central data administration group, the notion of "my data" will be maintained as opposed to the concept of "our data," which underlies any successful data base system endeavor.

The concepts of decentralized/distributed data base are often interwoven with the concepts of decentralized/distributed data processing. Thus far we have avoided a discussion of hardware advances and their impact on data base. Potentially the most disastrous occurrence would be unchecked proliferation of *independent* minis and micros throughout an organization. Decentralized data processing invariably leads to decentralized data bases, which invariably lead to the end of integrated data base systems.

To summarize, the hardware and software for distributed systems are quite far ahead of the organizational controls needed to support such systems. We may see a disastrous tangent in the development of decentralized data processing and data base systems.

Voice Recognition

Since 1990 is 10 years away, we must be permitted a little fantasizing. Science fiction devotees have long been familiar with talking, intelligent computers. HAL in the movie "2001" and the onboard computer on the starship Enterprise in "Star Trek" brought such systems to the attention of a large portion of the world's population in the 1970s.

How impractical are such systems? Again, the answer is surprising to many. Voice recognition is here — now, today. I have seen demonstrations of relatively powerful, although primitive, voice recognition systems that respond to numbers and arithmetic operators.

Can a voice recognition programming language be far behind? There is research going on now in this area.

Perhaps by 1990 Jean Smith from the ABC Co. will be able to press a button and ask for various pieces of data from the corporate data base.

Our job, over the next 10 years, will be to instill sufficient confidence in the data base to override Smith's temptation to send someone out to count the stock as a double check on the voice computer system.

The 1980s will be a period of maturity in DP. In data base, this maturity will be evidenced by the development of true data administration functions. Data and information systems will be recognized as important integral parts of an organization.

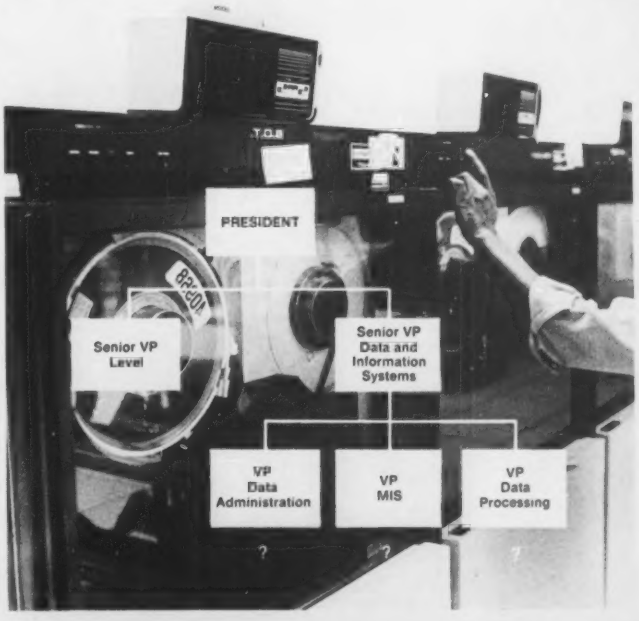
Technical advances in software will mainly be in the areas of query languages and data dictionaries. The latter will become integral parts of, or the drivers of, data base packages. Data base packages themselves will evolve to support relational structures. Software advances in the '80s will far outstrip our intelligent utilization of them.

Computer hardware will advance to a point where data base systems will be far easier to design and utilize than at present. Indeed, hardware advances will far outstrip software advances, which as already noted will far outstrip our ability to utilize them.

The rapid progress in hardware and software may actually result in decreased effectiveness of data base systems (a trend towards decentralization rather than distribution).

In summary, some organizations will be greatly pleased by the import of data base in the '80s. For most, however, it will be the late '80s before data base truly begins to demonstrate the potential predicted for it 10 years ago.□

FIGURE 2



INTERVIEW:

AT HOME WITH JAMES MARTIN

You are hypnotic on stage, but few people know what James Martin is like off stage. What do you believe in?

I believe that our generation is going to change the world. We've got a lot on our shoulders. Technology could make a world that's hell, and I think we appreciate that. We probably appreciate it much better than we did in the naive 1950s. Technology, on the other hand, could create a world that is just fabulous.

So I think about what my [four-year-old] daughter will be living in when she's my age. If we don't create something that's better than we have today, then we are pretty poor systems analysts — or we've done a lousy job of implementation.

So as a private individual I'm tied up in the things I talk about on my world seminars. I am just fascinated by technology. I live technology — not only the technology which I'm an expert on and write books on, but also I'm fascinated with technology on which I'm not so expert.

How do you relax?

I like to hike. I swim long distances; I sometimes swim in the sea for an hour or so the way other people jog. I enjoy photography and like portraiture very much. It is very fascinating trying to capture people with a camera. I am always surprised at how few people are good at portrait photography. I like Yousuf Karsh [the Ottawa-based portrait photographer]. I met him in an elevator once, and we chatted for a few minutes.

If you could choose to be any character of this century, who would it be?

I'd like to be a combination of Herman Kahn and Alfred Hitchcock. I've always wanted to be a fat man.

You are widely regarded as the foremost futurist in the data processing industry. Are you comfortable with that characterization?

Not entirely. I don't entirely like the word "futurist" because it sometimes has an odd connotation. There are a lot of cranks running around as "futurists."

But I do spend a lot of time trying to work out what telecommunications and other technologies are going to look like.

I think there are certain responsible statements you can make about the future and a lot of irresponsible statements. There are some statements we can make with a fairly high probability of them being correct. There are certain technology trends that are a fairly safe bet. Where you have a six-year lead time on a very complex project, that gives you a statement you can make about the future.

One can make speculative statements about the long-distance future, and I think it's important to do that because it's clear that electronic technology is going to change the world much more than the motor car did. Therefore, I think it's a responsibility of everybody who's making this happen to ask the question: What sort of a mess are we making? Or what can we create that's great?

You wrote in *Telecommunications and the Computer:* "There are forces for change of shattering magnitude inherent in today's technology. The public perceives them only dimly with a vague feeling of uneasiness." What are these forces and who is at fault for the public's dim perception of them?

Let's first talk about what the changes are. I think perhaps the most important one is that we're going to change almost everybody's job. A lot of this is going to happen in the 1980s because the 1980s is a decade which may combine turbulence with incredible developments in microchips and networks.

Another new industry that's growing very fast now — and growing much faster in Japan than in the U.S. — is the robot industry.

So if you look at networks, the office of the future, computer technology and the robot industry, there are not many jobs that are left which are not going to be affected.

Now if you change everybody's jobs, there is no question that a lot of people are going to be displaced from their jobs. And if one is socially responsible, one will want to adapt society so that they can get another different type of job so that you don't cause mass unemployment . . .

But rather mass displacement?

Yes, mass displacement, not mass unemployment.

Displacement is not necessarily bad because the jobs which we can automate tend to be the rather inhuman jobs — the routine paperwork, the routine work on the shop floor and so on. But as we automate these jobs it does mean that the people have got to move up to jobs which are more demanding. So they've got to learn more and they've got to be retrained. They will probably eventually earn more as the G.N.P. grows. And they will probably eventually have more leisure time. So the whole question about how people use their leisure time is a question that will become very important.

The combination of those factors means that we are changing society, and I suspect the rate of change in the next 50 years is going to be greater than the rate of change *ever* in any 50-year period in history. That change will be driven by technology — much of it electronic technology.

Of course there are a few other important technologies as well. Such as genetic engineering; hydroponics to grow more food; space technologies; new energy sources, with eventually nuclear fusion and solar energy.

As to the second part of your question, it is mainly the media that are at fault for the public's dim perception of these changes.

The most powerful medium by far is television, and television today is not by and large making responsible programs about technology and how technology is going to change people's lives. This is why I have an enormous desire to get hold of the television medium and do something about that. It's not a question of wanting necessarily to make any money out of it; I think it's important that the perception is put across to the general public.

One of the things we've got to say to them is, "You shouldn't be scared of the technology — it may be scary, but in the long run everything is probably going to turn out OK; with good things as well as bad things, and the good things will predominate. But on the way we're going to have to make a lot of changes which will be temporarily uncomfortable, like running out of petroleum."

Alvin Toffler, author of *Future Shock*, has a lot to say about the present and the future. In his latest book, *The Third Wave*, he writes about today's rebellion against runaway technology: "It involves a fast-growing army of people — by no means poor or unlettered — who are not necessarily antitechnological, or opposed to economic growth, but who see in the uncontrolled technological thrust a threat to themselves and to global survival." How do you differ from Toffler's assessment?

I don't differ from that assessment. Incidentally, I think that assessment is one-sided because he also says a lot of things in his book which are pro-technology. It is by and large not an antitechnology book. But it is saying that we want to change technology — technology which is usable by small companies, small entrepreneurs, small villages in India, rather than megalithic technology like gigantic nuclear power stations. He's advocating distributed technology in a sense.

Of course that's what's happening in the computer industry. Now we're getting distributed technology moving in pretty fast which is changing the responsibility of the end users and managers and changing the jobs of many people.

I wouldn't entirely disagree with Toffler's characterization of technology, but he may take it a little too far. One of the big questions for the future, in data processing *and* in society, is what ought to be centralized and what ought to be decentralized.

This is a continuing debate . . .

That's right, and I think it changes as the technology changes. For example, the fact that computing is now swinging more toward decentralized control is due to the technology rather than due to any pendulum effect. We're getting very cheap technology.

Is technology out of control?

"Out of control" is a word I would never use for the situation. I think man is in control, *but* there are certain aspects of it where you might use that phrase.

We're in the middle of a wild adventure, and the adventure is being driven by the expediency of the present time, whatever that time might be. So we don't know what the end of that adventure is going to be, and we don't have quite as much freedom of choice in selecting the end of the story as one would like to have.

Society is heading into the unknown pretty fast. So if one wanted to interpret that by saying that "technology is out of control," one could, if one was a sensationalist journalist, but I wouldn't use that phraseology.

Are you afraid of technology?

No, not at all. I think we are going to make mistakes with technology — big mistakes. But every computer programmer knows that he is going to write a program with bugs in it, and he also knows that he is going to be able to get the bugs out of the system. If Three Mile Island didn't have adequate safety controls, we know very well that 10 years later we are going to have power stations which have cor-

rected those mistakes. There are going to be other mistakes which are much bigger than Three Mile Island. But I think we are going to be able to recognize them and correct them — recognize them almost too late in some cases.

We make progress heuristically. Heuristic progress always leaves a few casualties along the route.

As hardware and memory continue to make quantum drops in cost, there is a strong tendency to erect ever more complex computer systems. Are these systems becoming too complex for any one person or group of persons to understand?

Yes, some of them are becoming too complex for one person.

Incidentally, it is going both ways. At the same time that we are building more complex systems, we are also building more simple systems.

How so?

The microcomputers, the small office systems, systems which are built of collections of micros.

If one looks at the best operating systems of the new minicomputer vendors and compares them with the biggest operating systems of the mainframe vendors, then you are achieving the same results and sometimes better results — like the increased reliability in Tandem computers, for example — in a much simpler fashion than the old operating systems.

So at the same time that there's a move toward increased complexity we are also seeing a move toward simplification.

There is a very old engineer in AT&T who is about 75 years old and retired, and he spent an evening once trying to pass on to me the experience of a lifetime. One of the things he kept coming back to, stressing over and over again, is that in the long run *great engineering is simple engineering*. And I think that is true with some of the systems we're building today. If we let them get so complicated that they're out of control or if we let them get too complicated for the system programmers to know what they're doing, then there is something wrong with the engineering.

Then are we on the right track to simpler systems?

I think both things are happening at the same time. Some systems are getting more complicated, some are getting too complicated.

Which ones are getting too complicated?

Any system where you haven't got the design tools to make it perform in a fully predictable fashion, for example.

There are two images one can create of distributed processing. One is where distributed processing becomes like a bowl of spaghetti, and every time you pull one piece it makes every other piece of spaghetti in the bowl shake.

The other image of distributed processing is a whole lot of modules, each one of which is largely self-contained. Each module is simple to understand by itself, each module has a standard interface to a network, the network is completely flexible so you can go from any point to any other point, the network is managed in its own right — it's well-managed, has good diagnostics, is reliable and highly flexible.

Here you have a system which could build up to enormous complexity and yet each component of it is understood and comprehensible. That's good engineering.

Are companies making adequate use of today's technology to process information?

I don't think they are. I think most companies have not yet perceived the full implications of computers for increasing productivity.

The reindustrialization of America has got to happen with much higher levels of automation and productivity, and concentration on the industries of the future, not the industries of the past. Productivity has slipped too much. And also productivity in data processing is nothing like as high as it should be.

I decided to swing my next round of seminars toward the subject of productivity because I think it's particularly important at the present time, especially if America is slipping into a recession.

What I'm doing at the moment is looking at every consideration which could increase productivity in data processing. The fascinating thing is, when you do that, the ones which most people are talking about are almost insignificant in their effect — like structured programming, structured analysis, structured design. They are *not* increasing productivity very much, whereas there are other forms of management of data processing that can make enormous step-function changes in productivity. We need to be asking the questions: What are they? Can we use them now?

My answer would be many of them can be used now, but there is a price to be paid and that is that you've got to change the management of data processing in order to use them now.

Is there a danger in what appears to be our over reliance on computers?

I think we've always got to keep it in balance. And if we build systems which could be knocked out by a catastrophic failure beyond our control, then there would be danger. It may be that some of the systems of society that we build today are too vulnerable.

Vulnerable to what?

Suppose for the sake of argument there were very bad economic times in the United States. I'm not predicting that there will be. But suppose that one of the largest banks in New York went belly up. There could be a whole chain reaction of events which might cause

---

The man generally acknowledged to be the computer industry's most widely read author, best attended lecturer and foremost authority on the social and commercial impact of computers greeted me at the doorstep of his home in an open sport shirt, tan shorts and bare feet. Tall and lanky, he towered over me as he towers over the information processing industry.

A graduate of Oxford University (with an M.A. in physics) and a British citizen, James Martin spent 19 years with IBM, both in the field and in development laboratories. For 10 of those years he worked at IBM's Systems Research Institute in New York.

Martin has discussed advanced technology on radio and television around the world and since 1977 has conducted the "James Martin World Seminar" on all six continents. He was cofounder and board chairman of Graphnet, America's first operational value-added common carrier.

Martin left IBM in 1978. He is now an independent consultant, a visiting fellow at Lancaster University in England, board chairman of The DMW Group, Inc. and vice-chairman of Butler Cox Associates, a London-based research group.

Among Martin's current projects is a study on the computing environment of 1985-1990 for one of the largest computer manufacturers. He is also working at the chairman level with one of the world's largest office equipment vendors.

Martin is an adviser to the British government. On the day before our meeting, he received a phone call at home from British Prime Minister Margaret Thatcher, asking him to consult on an overhaul of the British telecommunications industry.

Martin is only 41 years old.

I interviewed him at his secluded home in Bermuda, where he spends only 40% of his time, but where he does much of his writing. He is in the midst of his 23rd book (on data base), rising before dawn most days to work in his second-floor, book-lined "office of the future" overlooking the Atlantic Ocean. His latest book, The Wired Society, *was nominated for a Pulitzer Prize.*

Books are not Martin's only medium. He is also working with Deltak, Inc. to design and produce the Advanced Technology Library, a collection of more than 200 video-based courses on a wide range of information processing disciplines.

But Martin chiefly has "an enormous desire to get hold of the television medium." He has written a detailed synopsis for a proposed seven-hour television series about the future.

"The programs don't attempt to predict the future, but try to create a philosophy of what is now within our grasp and what could go wrong," Martin told me. "The last of the seven programs is a portrait of a utopian society which could possibly be created — but on the way there is all manner of chaos."

The series would cost about $7 million, and Martin is worried. "Chances are we won't ever get the money. The series probably won't be made. But the world absolutely needs it. Most of the programs made on technology are doomsday programs. Even the very good programs like 'Nova' have a negative tinge to them. This series says we could create a great world with technology IF. . . ."

Much of Martin's philosophy rests on that single word — if. He professes an unshakable belief in the power of technology to create that better world. And he is using every medium at his disposal to spread the word.

What he fears and mostly leaves unsaid is that man and his misuse of technology may foul things up.

Forceful and magnetic on stage, Martin in private conversation is polite, reserved, borderline shy. He speaks, as he writes, with simple eloquence. His carefully chosen words reflect intense thought, a vast knowledge of the technical literature and an intimate familiarity with the technological, economic and social forces at work in information processing and society at large.

Egotist? Dreamer? Genius? Martin has been called all three. But the lasting impression of James Martin is of a powerful man with a clear vision of the world as it is — and could be. If.

John Whitmarsh

the trucking industry to stop operating for two weeks, say. For two weeks there is not an adequate delivery system in the United States. What would happen if Manhattan ran out of food? What else would happen to American society? It would bring things to a screeching halt, and the results could be catastrophic.

This is perhaps too sensational an example. But you can look at certain examples of technology and ask the question, "What would happen *if* there was a certain improbable catastrophe?" If the answer to that question is that the whole damn thing is going to fall down like a pack of cards, then it's bad technology.

Are you also referring to vulnerability from terrorism?

We have not yet seen high-technology terrorism that is effective. I would think it's irresponsible to make that statement in a newspaper. If the terrorists read it, they might do their homework a little bit better.

New technology may be the factor which most changes the way corporations are managed over the next few years. What kind of MIS strategy should these companies assume to minimize the disruption of technology and also to take advantage of its opportunities?

That's a very complicated question and one which I happen to be writing a book on today.

There are quite a lot of components to it. A very important component is understanding what information is. So I'd like to use the phrase "information engineering."

Information engineering needs top management involvement in planning the information resources of a corporation. What we're seeing in many corporations today is the data processing department looking *out*. The data processing professional is never going to — in most cases — fully understand the needs of top management. He cannot know what a chief executive officer knows. He cannot know what the board of directors knows. Therefore, if information engineering proceeds from the data processing department outward, it is never going to be as good as if you use a methodology which involves top management in making certain statements about types of information which they feel are necessary to run the company.

What are now coming into existence are methodologies which enable one to look at information resources, translate those resources into data models, get the data models into a stable form that is independent of technology, translate the data models into today's data bases (which are going to change) and into distributed data bases, to work out the types of distribution you want on networks and *then* to select the tools which will give you the highest productivity in creating the types of information processing which a corporation needs.

How long does this process take?

It's not quick. There is a long cycle time associated with it which will vary from one corporation to another. So you won't be able to change the whole of data processing overnight. But if you haven't yet started, you are not serving your organization as you should.

But the first ingredient in "information engineering" is top management's involvement in the data processing function . . .

Yes, and in order to do that you've got to have a methodology. If you just go to top management and say, "Hey, you've got to get involved," top management will say, "Sure, I want to be involved. What do I do?"

At that point if you cannot give them a detailed action plan and hold their hand right through the plan, then it's not going to work. So you must have a methodology, and it's got to be a methodology which top management will get excited about.

But management seems as anxious as the general public about technology. They are afraid and don't really want to get involved . .

Yes, yes. One of the things I detect in the boardrooms at the present time is a very high nervousness about technology. Much more so than five years ago. One might almost say that some members of top management are almost scared of what is happening.

Why?

They have not got the subject in perspective. They don't know what to do about it. They don't know enough. They don't know the methodologies.

If somebody can go to them and say: "Do this! This is the way to protect yourself from the Japanese. Do this! This is the way to increase your productivity," then I think they would welcome that. I believe that top DP executives ought to be doing that.

A more aggressive stance with more options for top management?

Yes. But at the same time they've got to be careful how they do it. Because there have been a lot of mistakes in the last 15 years, and there is a substantial credibility gap between the DP executives and top management in many corporations (but not all).

Why the credibility gap?

I think we've oversold. We *badly* oversold MIS for top management. There are people who say now that MIS for top management will never work. I don't believe that at all. One can find certain examples of it that are working very well, but you've got to have an entirely different design methodology for creating it.

Does one particular company come to mind that has a top-notch MIS system?

"The reindustrialization of America has got to happen with much higher levels of automation and productivity, and concentration on industries of the future, not industries of the past."

CW Photos by J. Whitmarsh

"We're going to see much more dense memory and logic chips. We're certainly going to see 10M-byte chips, though that may take 10 years or more. That means the cost of main memory is going to plunge down. The military are asking for an IBM 370/158 on a single chip. I expect they will achieve that in the second half of the 1980s. The impact that will have on the possible range of products put on the market at low cost will be just gigantic."

"The most important thing in top DP executives is to be a businessperson. To think like a businessperson and to understand how the business works. To understand the businessman's vocabulary and needs."

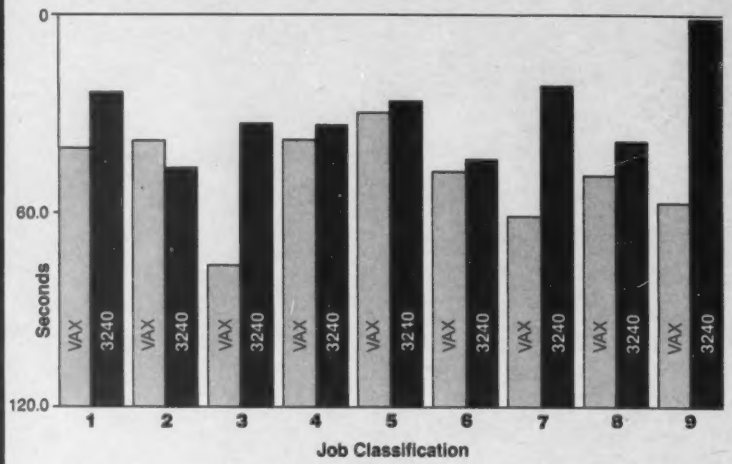
The benchmarks prove it...

1. Performance Measurements

Benchmark Name	VAX 11/780 CPU time	Perkin-Elmer 3240 CPU time	Performance Ratio*
PMS0154	1.140	0.907	1.26
PMS0254	4.710	4.485	1.05
PMS0354	9.160	8.990	1.02
PMS0454	43.610	43.451	1.00
PMS0554	86.420	86.123	1.00
PMS0664	1.610	1.350	1.19
PMS0764	7.040	6.679	1.05
PMS0864	13.730	13.370	1.03
PMS0964	66.830	66.068	1.01
PMS1064	132.920	129.727	1.02
PMS1124	1.520	1.306	1.16
PMS1234	2.470	2.424	1.02
PMS1324	1.310	1.618	0.81
PMS1407	251.560	37.327	6.74
PMS1507	50.260	57.657	0.87
PMS1674	22.440	23.005	0.98
PMS1724	52.550	53.133	0.99
PMS1807	50.180	34.766	1.44
PMS1907	32.020	34.679	0.92
PMS2007	17.720	20.264	0.87
PMS2107	13.450	20.242	0.66
PMS2202	33.311	26.352	1.26
PMS2324	121.610	137.997	0.88
PMS2402	0.940	0.539	1.74
PMS2502	0.720	0.562	1.28
PMS2624	122.920	227.689	0.54
PMS2708	4.850	0.961	5.05
PMS2809	103.770	0.006	17295.00
PMS2934	0.660	0.414	1.59
PMS3009	0.230	0.006	38.33
PMS3103	0.610	0.398	1.53
PMS3202	4.740	3.923	1.21
PMS3307	2.470	1.959	1.26
PMS3407	39.190	7.774	5.04
PMS3507	5.580	2.587	2.16
PMS3607	156.540	20.077	7.80
PMS3701	4.480	1.621	2.76
PMS3802	4.060	2.874	1.41
PMS3903	6.050	3.864	1.57
PMS4002	9.330	5.505	1.69
PMS4102	76.290	77.764	0.98
PMS4202	118.490	125.039	0.95
PMS4301	110.650	75.569	1.46
PMS4403	1.990	0.739	2.69
PMS4584	219.990	189.081	1.16
PMS4603	18.640	16.939	1.10
PMS4701	0.350	0.077	4.55
PMS4802	0.830	0.880	0.94
PMS4934	0.340	0.050	6.80
PMS5008	0.430	0.168	2.56
PMS5108	98.950	55.419	1.79
PMS5208	0.900	0.917	0.98
PMS5308	0.330	0.029	11.38
PMS5408	2.080	1.697	1.23
PMS5502	1.130	2.180	0.52
PMS5602	207.140	198.685	1.04
PMS5703	585.170	254.545	2.30
PMS5824	18.740	18.194	1.03

*Performance Ratio = $\frac{\text{VAX 11/780 CPU time}}{\text{Perkin-Elmer 3240 CPU time}}$

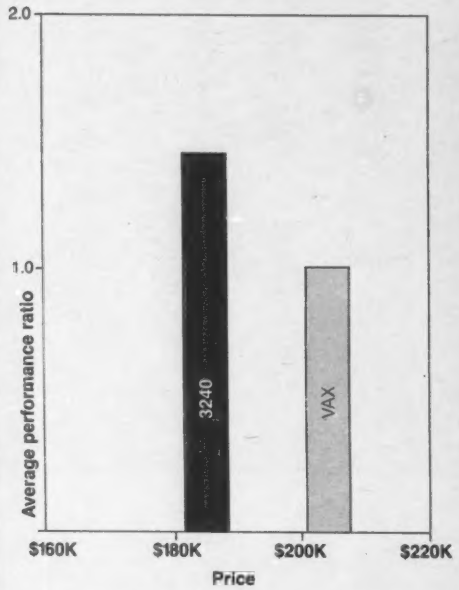
2. Perkin-Elmer 3240 vs. VAX* Average Time by Job Classification



1. Integer
2. Single-precision
3. Double-precision
4. Math functions
5. Single-precision whetstone timings
6. Double-precision whetstone timings
7. Arrays
8. Job mix
9. Compiler technology

*VAX is a trademark of Digital Equipment Corporation

3. Proven Price/Performance



The configurations tested were 2 MB of memory, 67 MB disc, 75 ips tape, Floating Point Processor, CRT terminal, battery backup, and system software. The 3240 had a Writeable Control Store and Fortran Enhancement Package, while the VAX 11/780 had a Floating Point Accelerator.

The Perkin-Elmer 3240 is Faster than VAX.

Our 32-bit Model 3240 performed 44 out of 58 benchmark tasks faster than the more expensive VAX 11/780.

We matched our Model 3240 against the VAX 11/780 in 58 customer supplied benchmarks, comprising 697 million FORTRAN statements.

In 44 of the benchmarks, the 3240 performed *faster* than VAX.

Overall, the 3240 delivered an average 1.4 times higher performance.

Yet the 3240 configuration costs approximately $20,000 less than its corresponding VAX configuration.

All 58 FORTRAN benchmarks listed in Table 1 were submitted by our customers and prospects reflecting their application needs. Applications that deal with the same kind of problems you face every day.

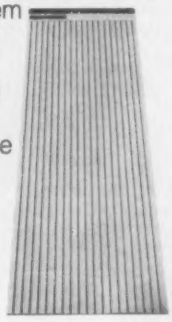
All benchmarks were run on-site at various locations across the country. And all were run by vendor-trained operators.

In Table 2, we organized the benchmark results into nine common types of job classification, so you can see at a glance how the 3240 stacks up vs. VAX 11/780. (The ninth classification demonstrates the superiority of Perkin-Elmer's system software, in this case globally optimizing FORTRAN VII.)

As shown in Table 3, the story is simple and uncomplicated. The Perkin-Elmer 3240 outperforms VAX 11/780. And it costs less.

The 3240 is faster than VAX in another important way—we can deliver it faster.

We'd be happy to send you a sub-set of those benchmarks which are non-proprietary or have been released with permission, so you can run them yourself. We'll also send you hardware and software documentation.



Before you make a decision on any supermini system, get all the facts on the new 3240 from Perkin-Elmer.

The Perkin-Elmer Corporation, Marketing Communications,
2 Crescent Place, Oceanport, NJ 07757

Please send me ☐ complete information on the 3240
☐ documentation ☐ a sub-set of benchmarks.
☐ Have a salesman call.

Name_____ Tel_____

Title_____

Company_____

Street_____

City_____ State_____ Zip_____

CW917

Call toll free: (800) 631-2154

In NJ: (201) 229-6800

PERKIN-ELMER

One sees a lot of different pieces of different types of MIS systems for different types of top management, which are working excellently.

Is MIS still an evolving function?

I think the word is used differently by different people. So "MIS" is a term which I am cautious about using.

In the work I'm now putting together, I'm dividing data processing up into four types of systems. One is a routine system. It processes volumes of work, very often a heavy-duty system which is completely predictable.

The second type of system is an information system. This is a system that doesn't change the information but makes information available where it's needed at the right time in the right level of quality and with the right type of presentation. We are now beginning to see much better information system technology.

The third type of system is a decision support system. This takes some particular type of decision and uses both information and algorithms for enabling that decision to be made much better.

The fourth system is a highly specialized system, like an operations research system or a design system.

In planning the data resources for a corporation, I think it's desirable to understand the different data needs of those four types of systems in planning your corporatewide structure of information resources.

What should the MIS structure be five years from now and how do we get there from here?

Traditional data processing has started with *processes*, and by using conventional systems analysis on the processes you then determine what files or data you need for operating those processes. If you look at today's structured analysis, it says examine the processes and examine the work flow, and the data will fall out as a by-product of doing that.

Now there is another technology which is almost the opposite to that. It says that we start by looking at the information needs, translate the information needs into data base structures, make the data base structures as stable as possible, then use high-level languages — much higher level than Cobol — for generating the processing on top of the data base with maximum productivity. That's almost exactly the opposite to the traditional way of doing processing.

Now one can translate that statement into a set of hard design methodologies for accomplishing it. It is that set of hard design methodologies that I think is going to be the mainstay of MIS five years from now. Much of what will be happening at that time will be the central or coordinated planning of the data structures, but those data structures being used in a highly decentralized fashion — by means of networks, distributed processing, minicomputers, small decision support systems as well as very large computers.

There is still a place for the very large computer?

Oh, yes. In fact, large computers aren't large enough today for the things we want to do with them. So the whole industry needs the next generation of Amdahl and the next generation of IBM very badly.

DP managers were once thought to be the next group to enter the executive suite, and some of them have. But top management today seems increasingly disgusted with DP. What will it take to improve the relationship?

I think the most important thing in top DP executives is to be a businessperson. To think like a businessperson to understand how the business works and and to understand the businessman's vocabulary and the businessman's needs.

At the same time, top management

Keeping pace with today's competitive business climate is a full-time, demanding job. The last thing your company needs under time-conscious decision-making pressure is a computer program that sometimes creates more problems than it solves.

So, the next time you have a business problem you want your computer to solve, try speaking your own language to it. English. And use a MODEL solution.

MODEL Formulas Equal Business Solutions.

MODEL is a versatile planning and modeling system that lets you use the English language to write applications. And helps your company make decisions fast.

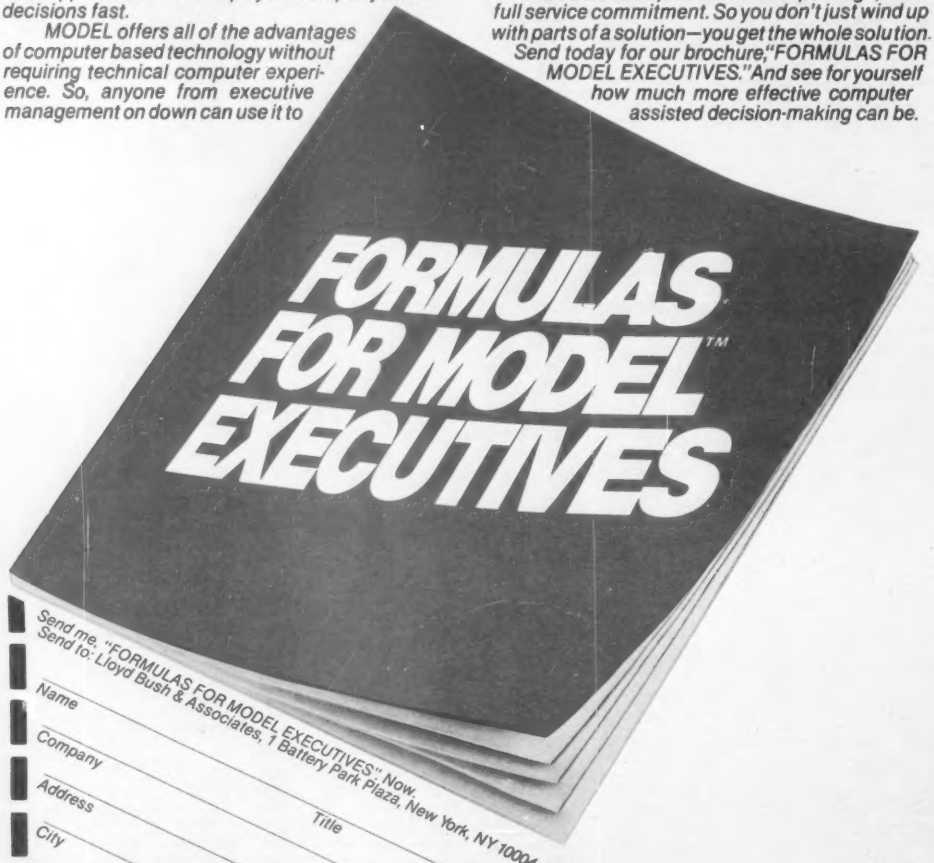
MODEL offers all of the advantages of computer based technology without requiring technical computer experience. So, anyone from executive management on down can use it to fit their needs. And all you have to do is let them know MODEL is available.

MODEL Formulas Multiply Your Effectiveness.

Every good executive has decision-making abilities. But the ability to make computer assisted decisions with speed, precision, and confidence is what makes a MODEL executive. And the advantages outlined in our free brochure will show you how you can multiply your company's effectiveness many times.

MODEL Formulas For Success.

MODEL isn't just a software package, it's a full service commitment. So you don't just wind up with parts of a solution—you get the whole solution. Send today for our brochure,"FORMULAS FOR MODEL EXECUTIVES."And see for yourself how much more effective computer assisted decision-making can be.



Send me, "FORMULAS FOR MODEL EXECUTIVES" Now.
Send to: Lloyd Bush & Associates, 1 Battery Park Plaza, New York, NY 10004

Name
Company
Address
Title
City
State
Telephone
Zip

See a MODEL demonstration by Lloyd Bush & Associates at INFO '80 New York Coliseum, October 6 – 9, 1980.

USE A MODEL™ SOLUTION.
LLOYD BUSH & ASSOCIATES

"Many of the statements which most people are making today are failures of imagination."

must understand that if it is going to be competitive by 1985, it's got to be operating a highly computerized corporation. And so it becomes the job of the CEO to make sure that a highly efficient computerized corporation comes into existence.

A lot of top management in the past has abdicated its responsibility for data processing; it has said, "This is something that can be done by the technicians." I don't believe it can be done by the technicians. The beginning of data processing has got to be a set of businessman's decisions.

But you said earlier that top management is afraid of technology . . .

I have just given a seminar in Australia for top management in which I didn't use many technical words. There is simply no need to tell them that intersection data in IMS can be handled by physical or virtual pairing with bidirectional pointers. You can phrase the whole thing in simple language which they can understand. This is part of a very basic communication process and that process is necessary if we're going to have good communication between DP people and top management.

Then Rule No. 1 is to get rid of the technical jargon . . .

When you are talking to non-DP people, you've got to get rid of the jargon. We obviously want to talk to ourselves in jargon because it's very useful shorthand. But we've got to understand that we do not use that shorthand when we're communicating with the board.

Unless you don't want them to hear you . . .

Yes, right.

You say that the job of data processing in the '80s is to build the computerized corporation. How?

By a set of information engineering methodologies. Ideally a methodology which enables top management to look at the objectives they want to accomplish. A methodology for the strategic planning of information resources in order to achieve those objectives. A hard methodology for translating those information resources into data models applicable to the corporation as a whole. Then a methodology that translates that into data bases. And the data bases unfortunately have got to incorporate the many systems which already exist today because you cannot convert them easily or cheaply. Network engineering that gives access to the data bases. Design techniques for distributed systems and for planning the distribution of data. Having created the data bases, then a set of high-level languages for doing everything we can to automate the application development process and automate the generation of information without having to write programs which are slow to write and expensive to maintain.

Now there are a lot of implications in what I've just said. One implication is that you're totally changing the job of the systems analyst. Another implication is that you're totally changing the job of most, but not all, programmers. You'd like to automate documentation, you *must drastically* lower the maintenance ratio.

Everything I'm describing is there, and you can see illustrations of bits and pieces of it. But if you look at almost any corporation, the statement you can make is that is hasn't fully understood it and hasn't fully got its act together. A corporation which is well-managed from a DP point of view ought to have its act together by 1985 in these areas — and the sooner the better.

Do corporations that are well-managed from the DP point of view have DPers at the vice-president level?

Absolutely.

In one of your other books you refer to Arthur C. Clarke, the science fiction author, who divides forecasting failures into two classes: failures of nerve and failures of imagination. You write: "Arthur Clarke's second category, failures of imagination, applies with a vengeance to the data processing and communication industries." What exactly did you mean?

Look at the 20-year history. Go back and look at what people were predicting 20 years ago. There were a lot of wild, crazy predictions — the computer-is-going-to-take-over-the-world types of predicitions.

But nobody 20 years ago would have predicted today's data bases, today's X.25 networks, today's microchips, the beginning of the industrial robotics industry. So all manner of things have happened which were not predicted.

Now, let's look at the next 20 years. If only it were possible for us to see the year 2000 now, I think what would be apparent to us is that many of the statements which most people are making today are failures of imagination. What is actually going to happen is going to be much more impressive in certain ways thanwhat most people are predicting today.

What I'm really saying is there are going to be technological surprises, but I don't think it's necessarily possible to describe them.

Suppose you take the subject of networking. The X.25 networks which are now spreading, by the year 2000 will go into absolutely every nook and cranny. They will be everywhere. Everywhere there will be inexpensive access to a network which moves data around the world fast and very, very cheaply. So all computers can be in communications at very low cost.

By the year 2000, just about everybody will have a pocket terminal. I think every school child will have one. It will be able to access computers inexpensively over these ubiquitous networks. The applications of that pocket terminal are endless because it is connected to computer systems, software banks and data banks everywhere by the networks and data banks.

By the year 2000 I would think just about every television set will be a computer terminal, like viewdata today. Viewdata will get very much better, but the cost of putting the viewdata chips into a television set soon is going to be very small — much less than the cost of the plastic case and knobs on the television set, probably. But in order for it to be interesting, you've got to have the networks and the data banks which become interesting for the mass public to use.

That would create vast security problems as well . . .

Yes, but there are solutions to the privacy and security problems.

Does this scenario include the replacement of printed newspapers and magazines by electronic versions?

I don't believe so. I think one of the aspects of an electronic society that is desirable is the maximum diversity of media.

I've done a lot of thinking because people have attacked me so many times about 1984 and totalitarianism and George Orwell and all that. I think the best way we have to protect ourselves from the possibility of future totalitarianism or future excessive control in government is to have the maximum diversity of information channels. So I don't want electronic newspapers to replace paper newspapers — they are a different medium.

If you have ever used an electronic newspaper (one or two of them do exist in Europe), it's totally different in feeling to the *New York Times* and the information you get is totally different. When you look through the *New York Times* you've got big pages and lots of long articles to read, you can scan through it by hand.

On the other hand, if I go to a viewdata set in London and ask what the *Economist* has to say about the Russian invasion of Afghanistan, in order to put that into electronic form they've had to structure it. So I've got a set of structured indices in which I can look up different aspects of the story. It is more structured than a story in the *New York Times*. The amount of read-

"The best way to protect ourselves against totalitarianism is to have the maximum diversity of information channels."

ing is smaller.

I may get insights from the *Economist* on viewdata that I do not get from the *New York Times*, but on the other hand I am going to get some things from the *New York Times* that I certainly don't get from the electronic magazine. I say they are different media so let's have both of them.

I do not believe that electronic newspapers will replace the *New York Times*. I think the *New York Times* and other such big papers will still be in business in the year 2000.

But only the marketplace will decide?

Only the marketplace will decide, yes. I think if you have free enterprise you probably will get maximum diversity of channels. But if it looks as though it's going to kill certain important types of channels, it may pay to have them artificially kept in existence. I think it's a very important principle that we ought to have maximum diversity of information sources in order to protect ourselves from the wrong forms of control of information.

If you look at television, for example, you might say isn't the best way to handle television to have absolutely free enterprise. But I think the public broadcasting service is clearly giving us something that's very important that we would not have got in the same time frame from free enterprise. I'd say the same thing about the BBC in Britain.

Should AT&T, which provides communications services, also provide the news and information that could be transmitted over those networks?

AT&T is just a common carrier. Are you suggesting that it should become a journalistic organization? I'd prefer to separate the common carrier business from the journalist business. If the two are put together, then with an organization the size of AT&T that becomes extremely powerful.

Considerable resistance is surfacing among professionals and executives in companies with state-of-the-art office automation systems. They complain that these systems impose their own structures and considerably narrow the freedom of old work habits. They say the systems are designed so that people must adapt to them instead of the machinery adapting to the people. They also claim they must make major psychological and behavioral changes under these systems. Are we moving too fast into the office of the future?

I don't think we're moving fast enough into any use of technology which increases productivity.

I think there's a much more uncomfortable danger ahead for America and for Europe than the danger which you've just mentioned, and that is the fact that the economy is going to be very sick. In order to correct the sickness of the economy we need to move as fast as we can into any area of improving productivity.

But having said that, you can certainly find lots of examples of systems which are badly designed from a psychological point of view.

Think how many training courses you could send a programmer on to learn the psychology of the terminal interface — there are very few such courses. I am just putting together a long training course on that subject with Deltak, and I think it will be absolutely unique. Having gone through the work of building this course with Deltak and writing my books on the subject, I am constantly amazed when I sit down in front of a new terminal system and play with it to discover the mistakes they have made. They are rather obvious mistakes which they would not have made if they had had the right sort of training. So the psychologist has an important role to play and that's becoming more and more recognized.

By some overly pessimistic estimates,

And turned it into a word processor.

Using our WP/34® software package, you can use your IBM System 34 for both data processing and word processing. Simultaneously. And do everything a stand-alone word processor can do at a fraction of the cost.

Once the system is in place (and it only takes an hour or so) you have a full-fledged word processor. Copy is displayed on a screen right in front of the typist and then can be manipulated in any number of ways. Words, phrases, lines and paragraphs can be added, moved or deleted. Headers can be put in. Pages can be numbered, automatically.

Once set up, error free letters and documents can be typed in a matter of seconds rather than minutes. A single page letter typically takes about 30 seconds.

If you have more than one terminal, they can all access your system concurrently.

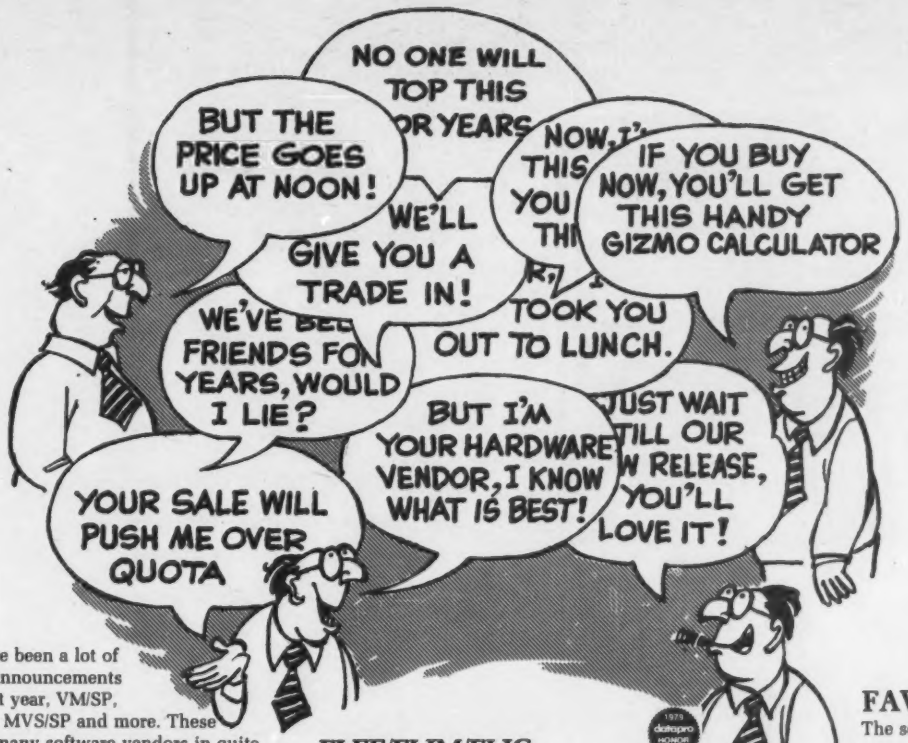
Printing can be done using an IBM ink jet or laser. And your typist can be setting up a second letter while the first is being printed.

If you own an IBM System 34 and are considering buying a stand-alone word processing system, it'll pay you to take a few minutes to talk to us. It could save you $10,000. Or more. Call our toll free number right now. 1-800-241-8291.

Financial Software, Inc.

2200 Northlake Parkway/Suite 270 Atlanta, Georgia 30084

HAVE I GOT A DEAL FOR YOU!


NO ONE WILL TOP THIS
BUT THE PRICE GOES UP AT NOON!
WE'LL GIVE YOU A TRADE IN!
IF YOU BUY NOW, YOU'LL GET THIS HANDY GIZMO CALCULATOR
TOOK YOU OUT TO LUNCH.
FRIENDS FOR YEARS, WOULD I LIE?
BUT I'M YOUR HARDWARE VENDOR, I KNOW WHAT IS BEST!
YOUR SALE WILL PUSH ME OVER QUOTA
JUST WAIT TILL OUR W RELEASE, YOU'LL LOVE IT!

There have been a lot of software announcements in the past year, VM/SP, DOS/VSE, MVS/SP and more. These have left many software vendors in quite a dilemma — no sales! Everyone wants to wait and see what the new software has to offer before making any decisions about what is already available. So a lot of software vendors have resorted to gimmicks to get you to make a decision to buy their product now. We're happy to say that Goal Systems hasn't had to use gimmicks. Our business policies (which have never changed since we were founded) allow you to make a decision now, but with complete freedom to change your mind.

About our free trial:
There is no financial obligation or absolute time restriction for evaluating any of Goal Systems products. We feel this policy is the best way to do business.

About our license:
There is no minimum license period for any Goal Systems software products. Any product may be used for as short a period as you deem useful.

FLEE/FLIM/FLIC

Put some integrity into managing your DOS/VS(E) libraries. Our 1500 users say this is the only LNKEDT, MAINT, and CORGZ they trust. Fast, flexible, and IBM compatible. $90/month-DOS/VS(E).

FAQS/FTL

This utility package combines console I/O spooling, very efficient partition balancing, and a transient manager and monitor. The result is a much improved operating system. $75/month-DOS/VS(E).

VFBA

VM DOS/VSE users can turn their old count-key-data disks into the new, high performance fixed-block-architecture disks without changing a single drive. You'll reduce CPU load due to I/O by 35%! $450 /month-VM.

FAVER

The sensible way to manage your VSAM backups. This easy to use, high speed VSAM dump/restore understands the intricacies of VSAM and puts some sanity back into safeguarding your valuable data. $80/month-OS/VS, $60/month-DOS/VS(E).

EXPLORE/DISCOVER

Next time you make a hardware decision, make sure you have all the facts with this easy to use, easy to understand software performance monitor. $110/month DOS/VS(E).

PHOENIX

The only manageable computer based education and training system available on IBM mainframes. Replace your unmanageable training methods with this truly unique system. $700/month-OS/VS.

To find out more, call one of our marketing representatives today.

Goal Systems Corp: The Goal-Getters.

1865 Fountain Sq./P.O. Box 29481/Columbus, OH 43229/(614)-268-1775/Telex: 245337

VAX Software excellence. Extended.

Digital introduces VAX/VMS™ software, Release II.

Since its introduction, VAX/VMS Performance Software has been setting new standards for excellence.

VAX/VMS was designed from the ground up by hardware and software engineers working together. The result: truly integrated computer software. Featuring highly efficient virtual memory, powerful I/O capabilities, and a basic operating system that's incredibly easy to work with.

Now with VAX/VMS Release II comes another major advance in VAX technology. With several new software capabilities added, and existing features enhanced, you have more ways than ever to use the power of VAX/VMS.

Powerful new languages.

With VAX/VMS Release II, our already remarkable FORTRAN is now a full ANSI 77 implementation and compiles twice as fast as before. We've also added COBOL and BASIC that compile at the rate of two to three thousand lines per minute.

The interactive BASIC gives you performance that approaches FORTRAN.

And COBOL performance on VAX is comparable to mainframe COBOL. Based on the ANSI 74 standard, it offers full support for: nucleus; sequential, relative and indexed I/O; segmentation; interprogram communication; table-handling; library; and sort/merge capabilities.

Other VAX languages include PASCAL, BLISS, and CORAL, the British real-time language standard.

But languages are only part of the story.

Advanced programming tools.

VAX/VMS gives you the ability to share commonly used subroutines in any language. And the ability to call any of the system services from any language.

In addition, there's an improved interactive editor that lets you create, proof and modify source programs right at the terminal. An interactive symbolic debugger that lets you debug your programs using source code statement numbers and symbolic names. And FMS for simplified screen formatting.

We've even enhanced the already easy-to-use Digital Command

Language by providing for user added commands.

Extensive data management.

To help you put all these programming tools to work, VAX/VMS Release II includes a wide range of data management facilities.

RMS, for example, allows you to set up sequential, random or multikey ISAM file structures in any VAX language. And you can use up to 255 keys for each entry.

Then to access RMS files you can use DATATRIEVE, a query and report writing utility that's both highly versatile and easy to use.

There's also a new VAX SORT/MERGE utility for easy record formatting.

State-of-the-art communications.

VAX/VMS is more than a powerful system in its own right. It also fits into any network or communications plans you may have.

Using DECnet, you can link VAX into a resource-sharing network with other computers from Digital. This network interface is transparent to programmers, which greatly simplifies your development work.

Then you can use 2780/3780 and MUX200 protocols to connect VAX to your mainframe system.

In addition, VAX/VMS Release II offers a new Mail utility for interterminal communications, even with a terminal on another CPU through DECnet.

And you can use the new DR780 hardware/software interface, with an unmatched 6Mb/sec throughput, to set up high-speed VAX-to-VAX communications or to support devices like array processors and graphics terminals.

Complete system control.

Best of all, VAX/VMS gives you complete control over system resources.

You can lock part or all of a program into main memory. You can set priorities on 32 different levels—the first 16 for real-time. You can establish disk, memory and processor quotas. You can even control user privileges to the point where it's virtually impossible for a low-level user to interfere with people doing high-level work.

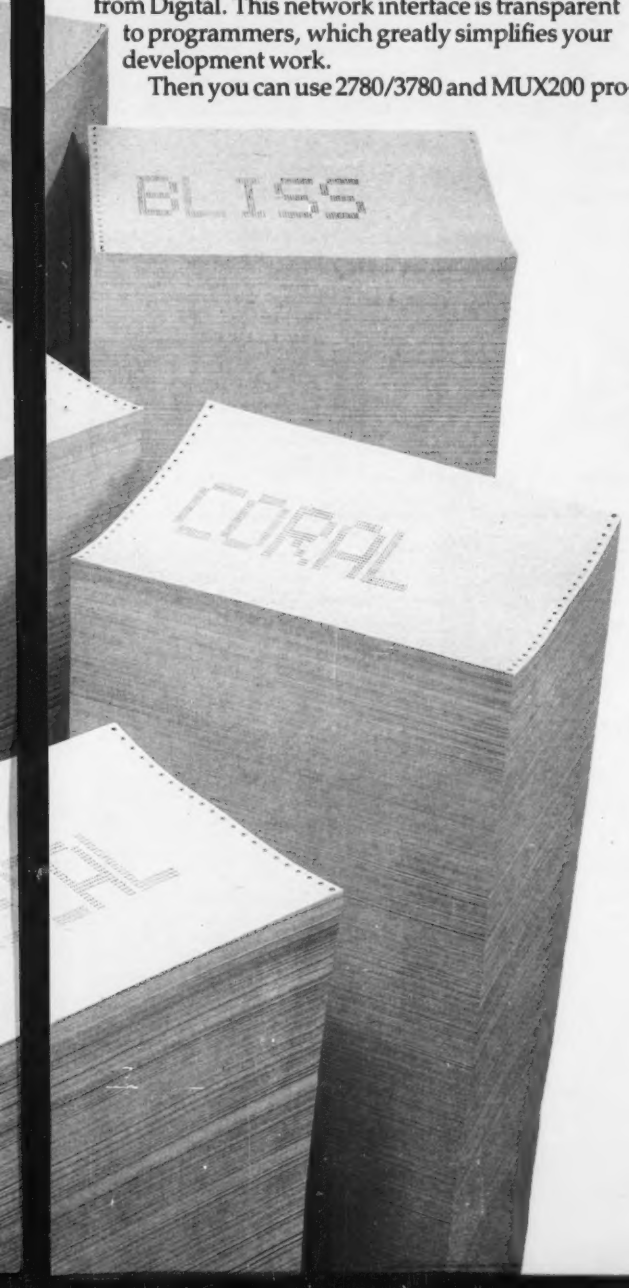
All this control lets you get the best possible performance from VAX/VMS. Regardless of your application or the number of people using the system.

The architecture of the '80s.

VAX/VMS Performance Software is part of a new kind of computer architecture. It combines the large program capacity of mainframes with the interactive access of minicomputers. It's completely integrated with VAX hardware for unprecedented system performance.

And now that Release II is here, all that power becomes available to a lot more people.

Just send the coupon for complete information.



I'd like to know more about the power of VAX/VMS Performance Software.

☐ Please send me your new brochure, "VAX Software: The Measure of Value".

☐ Please have a Sales Representative call.

Name_____

Title_____

Company_____

Address_____

City_____ State_____ Zip_____

My application is_____

Send to: **Digital Equipment Corporation,** 146 Main Street, Maynard, MA 01754, Attn: Communication Services, NR-2/2. Tel. (617) 467-6885. Digital Equipment Corporation International, 12 av. des Morgines, 1213 Petit-Lancy, Switzerland. In Canada: Digital Equipment of Canada, Ltd.

B-9-17-0

digital

THE COMPLETE PROJECT MANAGEMENT SOFTWARE SYSTEM

PAC II™

International Systems' PAC II Project Management System is a flexible multiple project scheduler, cost processor and manager, which respects resource limitations. It will help companies budget, plan, monitor, analyze, report and manage any type project.

PAC II's planning facilities support a simulator capability with critical path indications, resource loading, inter-project dependencies and full resource leveling. And the System has the capability to monitor your projects to help you keep them on target and within budget.

More than 600 installations worldwide attest to PAC's efficiency, accuracy and reliability.

Write or call for full details.

International Systems, Inc.

The world's leader in project management software.

890 Valley Forge Plaza, King of Prussia, PA 19406 (215) 265-1550

SUPERIOR SOFTWARE, INC.

SERIES/1

FINANCIAL CONTROLS SYSTEM

- *GENERAL LEDGER
- *ACCOUNTS PAYABLE
- *PAYROLL
- *JOB COST
- *PURCHASE ORDER
- *ACCOUNTS RECEIVABLE / INVOICING
- *ORDER ENTRY
- *INVENTORY MANAGEMENT
- *FIXED ASSET ACCOUNTING
- *SALES ANALYSIS

EDX-COBOL

AVAILABLE IMMEDIATELY

(800) 828-9106
(716) 664-3843

111 WEST SECOND ST.
JAMESTOWN, N.Y. 14701

We are PROUD

of our products...
of our people...
of our service...

and

YOU WILL BE PLEASED!

VIDEO DATA TERMINALS • MODEMS • DISC AND CASSETTE MEMORY DEVICES • TELEX TWX • 33, 40 & 43 TELEPRINTERS • CUSTOM PCBs • MINICOMPUTERS and the TELETICKETER®

SALES • LEASING • RENTALS • SERVICE • SUPPLIES

TERMINAL Systems, Inc.

11300 HARTLAND STREET, NORTH HOLLYWOOD, CALIFORNIA 91605
(213) 769-6772 OUTSIDE CALIFORNIA TOLL FREE 800-423-4248

DISASTER PLANNING SEMINARS

Learn how to design, test, and maintain an effective disaster recovery plan for your company. Our intensive, three-day workshops are taught by expert instructors and provide you with practical, hands-on experience. Call or write today for details.

Chicago	Sept. 15-17
Boston	Sept. 24-26
Atlanta	Oct. 22-24
Houston	Nov. 17-19
San Diego	Dec. 10-12
Key Biscayne	Jan. 21-23

EDP SECURITY, INC.

The contingency planning specialists

400-2 Totten Pond Road
Waltham, MA 02154
(617) 890-6466

---

innovative software for school administration

THE SCHOOL SYSTEM

- student scheduling
- marks reporting
- attendance
- transcripts
- report writer
- flexible data base design
- interactive and batch
- OS, DOS for IBM 370, 4300
- fully documented and maintained
- over 75 standard output reports

Contact:
Robert M. Delf
Vice President, Marketing
Columbia Computing Services, Inc.
8611 S. 212th Street
Kent, Washington 98031
(206) 872-0283

PMS: The online interactive PERSONNEL MANAGEMENT SYSTEM for complete management and work flow control of personnel and employee benefit activities.

For information or a demonstration contact:

TESSERACT CORPORATION

101 Howard Street
PO Box 7658, San Francisco, CA 94120
Phone: 415-543-9320

"The industry is absolutely forced to change the methodology of creating applications within the next five years."

from 50% to 75% of all U.S. factory workers could be replaced by smart robots by the end of the century. Office workers, too, wonder if word procesing and automation will replace them even sooner. Is a confrontation between labor and management over automation inevitable at the bargaining table? What is the possible outcome of such a confrontation?

We're certainly going to have a confrontation at the bargaining table. The question is, how bad does it become?

I've recently spent some time with some of the trade union leaders on the subject. This came about through another film I'm making with Deltak called "The Microelectronics Explosion." One of the trade union leaders that we had in the studio was Thomas Weekly, who is the United Auto Workers person in charge of activities relating to robots. As you know, the automotive industry is considering robots very heavily at the present time.

Weekly's attitude is that in the long run it will be good for the auto worker if robots are introduced *extensively*, on a wide scale. The reason for that is that it will enable a person who is doing a rather boring and rather inhuman job today to move up to a job which needs more training, is more interesting to do and will give him higher pay.

So I think the attitude of the most responsibie union leaders is that in the long run they very definitely want these things to happen. In the long run they are going to increase the wealth of everybody.

You said recently that "structured programming increases productivity only about 10%. It is *absolutely* not the answer. We need far more powerful tools with far greater orders of magnitude."

That's right. If you look at the rate of dropping cost of computers, we know fairly well what chips are going to exist by 1985. So we can project the cost of computers by 1985. On that basis you can try to work out how many programmers you would need to program them *if* you did it with today's productivity. The answer to that question is ridiculous.

The industry is absolutely forced — it has no option — to change the methodology of creating applications with a step-function change within the next five years. Otherwise it simply won't be able to sell the computers that it is creating.

If you look at the amount of application programming, systems programming and systems analysis that is going on with the average IBM 4300 at the moment, it's totally out of balance with the cost of the machine. If you then think that that machine is going to be a very small fraction of that cost by 1985, then you see my point.

Do I understand that you believe the 4300 is underutilized right now?

I'm not saying it's underutilized. I'm saying the cost of getting it to work is too much compared to the cost of the machine.

Yet you said earlier that we need far more powerful mainframes . . .

Yes, very definitely.

. . . And far more powerful tools to increase efficient use of these larger machines?

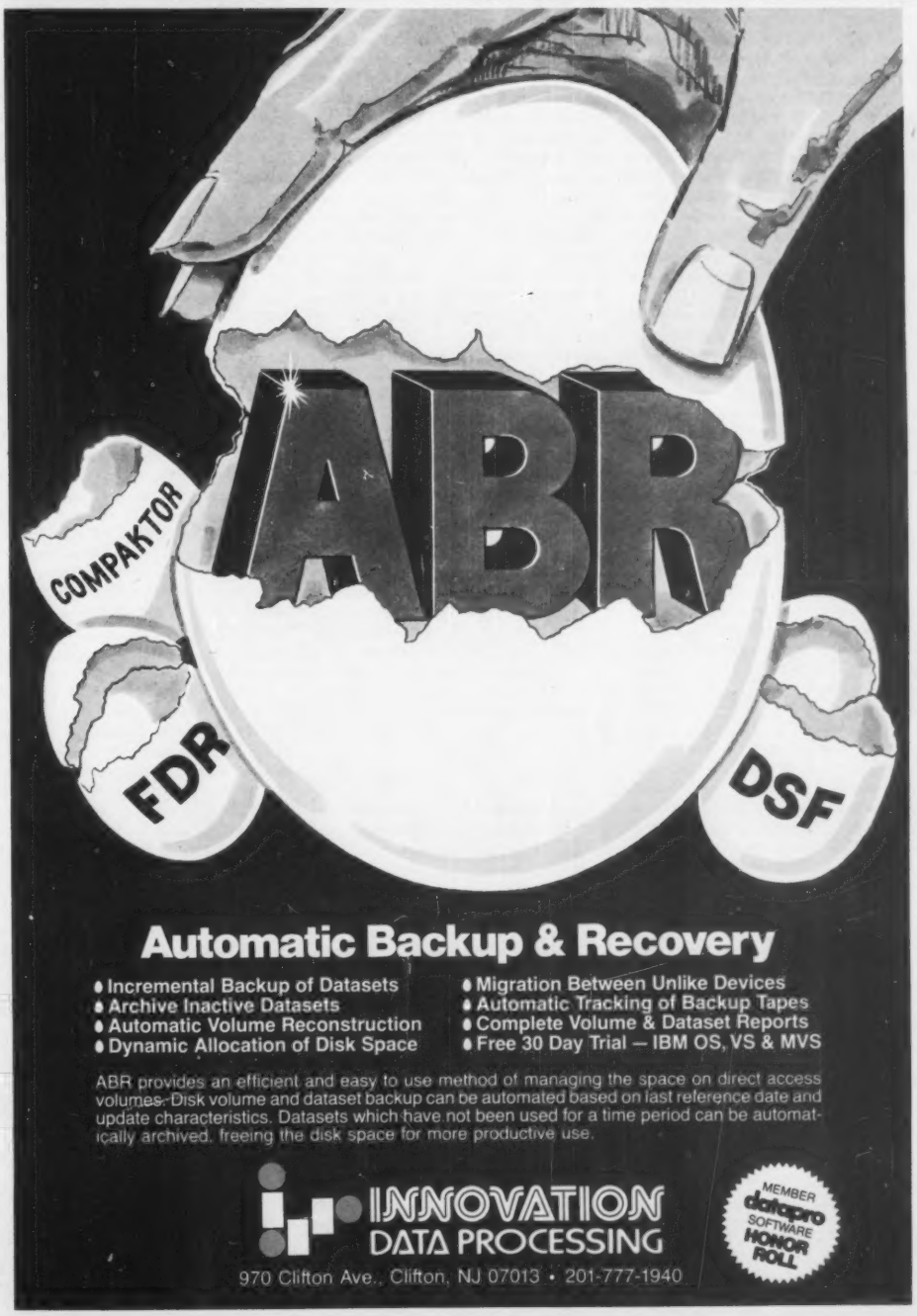
Yes, absolutely.

Is automated programming among those tools?

I don't normally use the phrase "automated programming."

There are a variety of different types of new languages. Many of them relate to data bases. In fact, these new languages are more powerful if the machine already understands the data and if the data is described in the dictionary that is built into the system.

(Continued on Page 113)


ABR
COMPAKTOR
FDR
DSF
Automatic Backup & Recovery
• Incremental Backup of Datasets
• Archive Inactive Datasets
• Automatic Volume Reconstruction
• Dynamic Allocation of Disk Space
• Migration Between Unlike Devices
• Automatic Tracking of Backup Tapes
• Complete Volume & Dataset Reports
• Free 30 Day Trial — IBM OS, VS & MVS
ABR provides an efficient and easy to use method of managing the space on direct access volumes. Disk volume and dataset backup can be automated based on last reference date and update characteristics. Datasets which have not been used for a time period can be automatically archived, freeing the disk space for more productive use.
INNOVATION DATA PROCESSING
970 Clifton Ave., Clifton, NJ 07013 • 201-777-1940
MEMBER
dataPro
SOFTWARE
HONOR
ROLL

By Herbert L. Gepner

APPLICATIONS SOFTWARE EXPLOSION

The use of applications software packages got off to a slow start in the 1960s, but has gained steady momentum since. The author surveys the present status and future prospects of "off-the-shelf" packages, supplies some guidelines for purchasing them and concludes with a list of 100 important software vendors.

Today's demand for and availability of virtually every conceivable type of applications software transcends all computer lines, from the very small to the very large. There are software vendors which offer simple individual modules that can be executed on small business computers, either sequentially or as part of an integrated system. At the same time, there are multiphased, totally self-contained manufacturing systems developed by both hardware manufacturers and independent software suppliers for use on large mainframes. Developers of data base systems have begun to realize the benefits that can be derived from supplementing their systems with interfacing applications software.

Where does the applications software industry stand as we enter the decade of the '80s? The minicomputer/-microcomputer marketplace is now the dominating factor in terms of the volume of software development. An important factor to be recognized in this new information revolution is that the software being designed for these small systems is "user-dictated," as opposed to the early mainframe software, which was largely vendor-dictated.

It should also be recognized that much of the microcomputer applications software being developed is very simplistic in nature, to complement the low-cost hardware for which it is being designed. (This is not to say that there is no sophisticated software for these small systems. Many of the software systems designed to run on larger mainframes, such as data base management systems and telecommunications monitors, have been modified to fit into the smaller processing units.)

New types of applications are being demanded by the users of these inexpensive, versatile systems. The products now pouring onto the market in response to these user demands include graphic design systems that will prepare management reports without secretarial assistance and do it at the moment of request; interactive design support packages for architects and automobile designers; hotel and motel reservation systems; doctor and lawyer patient/client accounting systems; and many other unique applications.

'Caveat Emptor'

The minicomputer and microcomputer manufacturers cannot hope to satisfy this large and diverse demand for applications software, so they are turning to the individual software developers and acquiring the marketing rights to their packages. In many cases, they pay an acquisition price and establish some sort of royalty plan for the developer. Occasionally, they purchase the packages outright.

The end-user purchaser of a microcomputer software package should be aware of the risks involved in acquiring some of these packages. With mainframe software, a stipulated warranty or guarantee usually goes along with the sale. With microcomputer software offerings, the watchword is "caveat emptor," since more often than not the package carries little, if any, guarantee that it will do what the user wants it to do.

The risk, however, is not as great as one might imagine because of the previously noted simplistic nature of most of this new software. When a low-priced package is designed to do one job and do it well, that is all the user should demand. This, however, requires the user to know what he really wants.

We certainly don't mean to imply that all the applications software activity is taking place in the small-system arena. The availability of software for the larger mainframe computers is growing dramatically in the fields of data base management, communications and large applications systems such as total manufacturing systems. Whereas the small computers lend themselves to straightforward software packages for individual applications, the larger mainframes are geared to handle large banks of data with sophisticated, interfacing applications and systems software.

What the Future Holds

Now let's polish our crystal ball and look at the applications software of the future. In the table of contents of the *Datapro Directory of Software*, we have identified 12 specific applications areas that are prominent in today's marketplace: accounting; banking and finance; education; engineering and scientific; insurance; management sciences; manufacturing; mathematics and statistics; medical and health care; payroll and personnel; sales and distribution; and word processing.

This list is by no means exhaustive, and each of the 12 categories can readily be further subdivided, as we have done in the *Datapro Directory*. Moreover, certain types of software transcend several categories and therefore are more difficult to pigeonhole.

These 12 major applications areas, along with the new technologies brought on by the word processing and automated office revolution, the expected growth of distributed processing architecture, the increased importance of international satellite communications networks and — most of all — the proliferating mini/micro systems will dominate the applications software industry of the '80s. Let's review the anticipated future trends in each of these applications areas.

Financial Software

In the 1980s we will see a steady and continued growth in the demand for more extensive and effective accounting packages to meet the growing paperwork requirements at both the federal and state levels. Payroll and personnel systems will continue to be in high demand, especially as the payroll activities within groups of businesses become increasingly standardized (again, thanks to government regulations).

The growth in these areas will be strongest in the mini/micro sector. Only slow to average growth in new accounting and payroll applications software for large systems will be noticed, and most of this work will be a result of the ever-increasing government requirements. The established vendors servicing the large mainframes will continue to enhance their existing offerings and to combine interacting modules (or packages) into large data base-oriented systems.

We do not want to infer that the development of applications software for large computers has bottomed out. To the contrary, as mentioned before, certain major applications systems are only beginning to make their presence felt in the marketplace.

There is a growing demand for manufacturing control systems and the next decade will bring major advances in this area. Project planning and standards development systems will also come to the fore. After all, now that computers are making mountains of data available to management, there have to be better ways to control and organize this data so management can derive meaningful information from the ever-increasing data bases.

Banking software will again undergo a flurry of activity, especially in response to the demands that will be generated by federal action regulating electronic funds transfer and by the increasing use of automated teller machines. In addition, we foresee an increasing demand for sophisticated financial planning and control software, especially in the area of protfolio management.

Medical Systems

One of the more promising applications areas, medical and health care, will continue to be relatively slow in evolving, but only in the proprietary software marketplace. Most of the systems and development work in this area is currently being undertaken by contract software houses, tailoring special systems to satisfy individual hospital environments.

Hospitals not big enough to afford their own systems are typically using service bureau-type facilities offered on a usage basis. Several large remote network service vendors are installing hospital-oriented services on their systems to meet the growing demand.

On the other hand, small systems for doctors' offices and small clinics will gain widespread acceptance when the mini/micro market calms down and doctors realize the benefits afforded by in-house recordkeeping. Although mini/micro packages for doctor/clinic accounting will be available in large numbers, total hospital administration systems available "off-the-shelf" will continue to be relatively scarce. We also predict relatively slow growth in customized systems for laboratory control.

Education Slow, Too

Education is another area in which we foresee slow growth. Most educational institutions, except for a few well-to-do universities, are hard-pressed to afford the expenditures required to install high-powered computer systems. The installed computers are being used mainly for school administration functions.

Many school districts share computer services on their local community systems or go outside to independent service bureaus. As we progress into the financially strapped '80s, there does not seem to be any relief in sight.

Despite this gloomy outlook, significant strides have been made in the field of computer-aided instruction (CAI). CAI has evolved out of combined manufacturer/community/government experimental efforts. The benefits have been widely observed and reported. Dramatic results have been noted, especially in inner city areas where traditional teaching of the learning skills has long been a frustrating problem.

However, unless a dramatic influx of funding is forthcoming, it will be many more years before this promising technology will be fully developed and widely available.

Scientific Software

Software addressing the engineering, mathematical, statistical and other scientific areas will continue to evolve along with the underlying technologies. Since the beginning of the computer age, these application fields have been of primary concern to scientists of all disciplines. We see no letdown in the '80s, especially in light of the de-

Herbert L. Gepner is group managing editor/software at Datapro Research Corp., Delran, N.J.

mands for computer solutions to our society's increasingly complex social and economic problems.

One area that is certain to attract a great deal of attention in this decade is energy. This will be closely followed by the increasing efforts to protect the environment in which we live.

If there were no other reasons for predicting marked growth in the engineering/scientific areas (and there are many), these two problems alone would challenge the scientific software developers.

Sales and Distribution

We predict that sales and distribution software will experience moderate growth during the '80s, with more activity in the distribution area than in the sales area. Even though retail sales technology attracted a great deal of attention during the '70s, the acceptance of computerized applications has been rather slow, and the dismal near-term economic forecasts provide no reason to be optimistic about any sudden upsurge in demand for retail-oriented software. Most new developments in this area will probably come off the hardware manufacturers' drawing boards.

Automated systems to speed the distribution of products will be more in demand as a result of the desire to conserve energy and establish better control of the movement of goods from the warehouse to the dealer. As energy becomes scarcer, we can expect to see more large companies employing automated inventory control and distribution systems, and they will look first toward the proprietary software market to meet these needs.

Word processing (WP) and automated office system software deserves special mention. As noted before, we are entering an era frequently referred to as the "information revolution." A quick look at what is happening in the WP market validates the appropriateness of this term. More and more offices are converting to WP to increase the efficiency and accuracy of their operations; ordinary typewriters are being replaced with WP systems at an astounding rate.

The 1980s will see the marriage of WP systems and mini/microcomputers consummated, and one of the results of that union will be a great demand for sophisticated text editing and WP software. Initially, we can look to WP and computer equipment manufacturers to lead the way in supplying this software. The independents will soon follow suit and enter this marketplace, with initial emphasis on the most popular computing systems, such as the Digital Equipment Corp. PDP-11.

Some independents have already taken this step. Ultimately, however, when the inevitable shakeout occurs, the major suppliers of this type of software will be the successful manufacturers of WP systems.

A specialized branch of WP, electronic mail, is another area where much is being said but little is actually being done. The major efforts are currently being directed toward in-house development and usage by very large companies, including computer manufacturers. This technology is slowly making its mark in the larger commercial remote processing networks as well.

Although the '80s will see more activity in the evolution of electronic mail technology, actual in-house installation of these systems will be slow and selective.

Accent on Micros

The big thrust in the '80s, as noted before, will be applications software for mini/micro users. More and more people are getting involved with personal computing hardware and software. Packages to help prepare income taxes, keep track of household budgets, prepare account records and play nearly every imaginable kind of game are available for as little as $9.99.

Is this a significant trend for software development in the future? You'd better believe it! Just look in any shopping mall for the nearest Radio Shack store and judge for yourself. Go to any of the proliferating personal computing shows and view the thousands of enthusiasts who come to buy.

The 1980s have begun with a mini/micro marketplace that is noteworthy for its lack of organization. In fact, a better characterization might be "cha-

---

The following is an alphabetical list of 100 well-known applications software vendors. This list includes vendors whose products were either mentioned in Datapro Research Corp.'s 1980 survey of proprietary software users or identified through subscriber inquiries to the Datapro Consulting Department.

It's important to note that there are hundreds of other respectable vendors of applications software. The *Datapro Directory of Software* and the *Datapro Applications Software Solutions* reference services list far more vendors and products than can be listed here.

In most cases, the field(s) of applications specialty are noted below each vendor's name and address. In the case of the major computer manufacturers, however, the specialty fields have been omitted because of the wide range of offerings available.

American Valuation Consultants, Inc.
Suite 247
2200 E. Devon Ave.
Des Plaines, IL 60018
General Accounting
Medical Administration

AM/Jacquard Systems
3340 Ocean Park
Santa Monica, CA 90404
General Accounting
Payroll
Word Processing

Argonaut Information Systems, Inc.
Suite 6
383 Grand Ave.
Oakland, CA 94610
General Accounting
Payroll

Aries Information Systems, Inc.
Suite 370
7400 Metro Blvd.
Minneapolis, MN 55435
General Accounting
Payroll

Arthur Andersen & Co.
69 W. Washington St.
Chicago, IL 60602
General Accounting
Manufacturing
Medical and Health Care
Sales and Distribution

ASK Computer Services, Inc.
730 Distel Drive
Los Altos, CA 94022
General Accounting
Manufacturing

Atlantic Software, Inc.
Layfayette Bldg., Suite 910
5th and Chestnut Sts.
Philadelphia, PA 19106
Management Sciences

The Automated Quill, Inc.
Suite 5
3501 S. Corona St.
Englewood, CO 80110
General Accounting
Construction Accounting
Manufacturing

Bancroft Computer Systems, Inc.
715 Trenton St.
P.O. Box 1533
West Monroe, LA 71291
General Accounting
Graphics
Manufacturing
Payroll
Word Processing

Basic/Four Corp.
14101 Myford Road
Tustin, CA 92680
Management Sciences
Medical and Health Care

Baybanks Data Services, Inc.
235 Wayman St.
Waltham, MA 02154
Banking and Finance
Payroll

M. Bryce & Associates, Inc.
1246 Springfield Pike
Cincinnati, OH 45215
Performance Standards

Burroughs Corp.
Burroughs Place
Detroit, MI 48232

CACI
12011 San Vincente Blvd.
Los Angeles, CA 90049
Management Sciences

Cado Systems Corp.
2771 Toledo St.
Torrance, CA 90503
Auto Sales Accounting
Insurance

California Computer Products, Inc.
2411 W. LaPalma Ave.
Anaheim, CA 92801
Graphic Arts

Chi Corp.
11000 Cedar Ave.
Cleveland, OH 44106
Mathematics and Statistics

Cincom Systems, Inc.
2300 Montana Ave.
Cincinnati, OH 45211
Manufacturing

Computer Corp. of America
575 Technology Sq.
Cambridge, MA 02139
Electronic Mail

Comserv Corp.
Suite 211
3050 Metro Drive
Minneapolis, MN 55420
General Accounting
Manufacturing
Sales and Distribution

Condata, inc.
1809 Walnut St.
Philadelphia, PA 19103
Payroll/Personnel

The Continuum Co.
3429 Executive Center Drive
Asutin, TX 78731
Insurance
Medical and Health Care

Control Data Corp.
8100 34 Ave., South
P.O. Box 0
Minneapolis, MN 55440

Cullinane Corp.
20 William St.
Wellesley, MA 02181
Auditing

Cyborg Systems, Inc.
Suite 2225
2 N. Riverside Plaza
Chicago, IL 60606
Payroll/Personnel

Data Design Associates, Inc.
Suite 418
1333 Lawrence Expressway
Santa Clara, CA 95051
General Accounting

Data General Corp.
4400 Computer Drive
Westboro, MA 01580

Datapoint Corp.
4204 Gardendale Road
San Antonio, TX 78284

Digital Equipment Corp.
146 Main St.
Maynard, MA 01754

Disc, Inc.
(Formerly Hittman Associates)
110 Painters, Mill Road
Owings Mills, MD 21117
General Accounting
Banking and Finance

Execucom Systems Corp.
P.O. Box 9758
Austin, TX 78731
Financial Planning

Financial Systems Consultants Division of Indiana Bank
915 S. Clinton St.
Fort Wayne, IN 46802
Banking and Finance

Financial Technology, Inc.
612 N. Michigan Ave.
Chicago, IL 60611
Banking and Finance

Finar Systems Ltd.
Suite 212
132 Nassau St.
New York, NY 10038
Financial Analysis

First National Bank of Boston
60 Walnut St.
Wellesley, MA 02181
Stock Purchase

Florida Software Services, Inc.
P.O. Box 2269
Orlando, FL 32802
General Banking and Finance
Manufacturing
Payroll

Fortex Data Corp.
10 South Riverside Plaza
Chicago, IL 60606
General Accounting

Harris-SAI, Inc.
611 Church St.
Ann Arbor, MI 48104
Electrical Engineering

Haverly Systems, Inc.
78 Broadway
Denville, NJ 07834
Management Sciences
Mathematics and Statistics

otic." Hardware is coming from hundreds of sources, and software seems to be coming out of every backroom or kitchen that can house a TRS-80, Apple or Pet microcomputer.

As in the mainframe market, where order and direction was established when IBM introduced its third-generation computers and proceeded to lead the development of appropriate software, we can look forward to IBM and a few other major manufacturers, such as DEC and Data General Corp., to lead the way toward an organized mini/micro market in the '80s.

We believe this transition can take place within as small a time span as five years. By that time, a few leading mini/micro software houses will have clearly risen above all the rest. Among these will be companies such as Microsoft, Inc., producing compilers for virtually every type of microprocessor on the market; Personal Software, with its business application packages; and Digital Research, Inc., with its operating systems and communications control software.

Going out on a limb, we project that there will be 10 major microcomputer software suppliers when the market finally settles down.

One factor that will take on extreme importance in the small business computer market is software portability. The ability to acquire or develop an application and run it on a variety of hardware systems will be a "must" in this industry. As he moves back and forth from 16-bit to 32-bit processors, the user should not have to change his applications software.

One approach to solving the software portability problem has been introduced by Apple Computer, Inc. for its 6502-type microprocessor. The user can purchase, for approximately $400, the "softcard," actually a separate microprocessor that enables the Apple to "emulate" the Z80 or 8080 environment.

Considering that there are now more than 200,000 users of Radio Shack's Z80-based TRS-80 microcomputer alone, it's easy to understand the benefits afforded by this type of portability.

Firmware Advances Key

Firmware advances will play a large role in the mini/micro market. Mainframes users will also benefit from a steady movement toward firmware in the '80s. Segments of operating systems, access methods, I/O routines and even data base management systems will be increasingly implemented in firmware for increased efficiency. In fact, almost any coding that can be depended upon to be static in its function is a candidate for firmware implementation.

The serious small businessperson will be increasingly likely to install a mini or micro system of his own, especially if his is a small establishment that doesn't need a complex, multiphased applications system. This potential computer user has two ways of proceeding: He can acquire his own computer system from the manufacturer and purchase the necessary software to handle his operations, or he can go to a turnkey vendor and have him install a ready-to-use system.

The turnkey vendors have been and will continue to be an important factor in the growth of the applications software industry. These innovative firms purchase the hardware modules that make up a minicomputer or microcomputer system and develop specialized industry software to run on that system.

At first, the turnkey vendors concentrated on the lucrative, never-ending demand for accounting software systems. One of the major developers of minicomputer accounting software is MCBA, whose offerings are constantly being modified to support a wide variety of computers. Turnkey vendors soon broadened their efforts to include specialized systems for the pharmaceutical, hotel and auto industries, and today you can probably find a turnkey vendor addressing almost any specialized application for which there is a measurable demand.

It's also worth noting that the turnkey vendors frequently offer their software packages "off-the-shelf" to users with compatible hardware systems already installed.

As systems become easier to program and the accompanying software becomes simpler in nature, the turnkey industry will probably level off. However, for the next five years at least, it should enjoy vastly increased revenues.

Glowing Prospects

We can conclude that applications software in the '80s will see sizable growth in acceptance and revenues in most segments of the business and scientific fields, with truly spectacular growth in software for the small business and personal computing environments. Although the revenues produced by individual applications products may not approach those produced by large systems software products because of the wide differences in product prices (e.g., a mainframe data base management system may sell for $150,000, while an accounts payable package for the same computer may sell for $15,000), the installation frequency of applications software products will greatly exceed that of the systems software offerings.

As this trend continues, we expect to

HBO & Co.
Suite 111
219 Perimeter Center Parkway N.E.
Atlanta, GA 30346
Medical and Health Care

Hewlett-Packard Co.
1501 Page Mill Road
Palo Alto, CA 94304

Honeywell, Inc.
200 Smith St.
Waltham, MA 02154

IBM Data Processing Division
1133 Westchester Ave.
White Plains, NY 10604

IBM General Systems Division
5775 Glenridge Drive N.E.
Atlanta, GA 30301

Infonational
P.O. Box 82477
San Diego, CA 92138
General Accounting

Informatics, Inc.
21050 Vanowen St.
Canoga Park, CA 91304
General Accounting
Banking and Finance
Insurance
Manufacturing
Payroll/Personnel

Information Associates, Inc.
97 Humboldt St.
Rochester, NY 14609
General Accounting
Education

Information Science, Inc. (Insci)
95 Chestnut Ridge Road
Montvale, NJ 07645
Payroll/Personnel

Insurance Systems of America, Inc.
P.O. Box 47975
Atlanta, GA 30362
Banking and Finance
Insurance

International Mathematical & Statistical Libraries, Inc. (IMSL)
Sixth Floor
7500 Bellaire Blvd.
Houston, TX 77036
Mathematics and Statistics

International Systems, Inc.
890 Valley Forge Plaza
King of Prussia, PA 19406
Banking
Management Sciences

K&H Computer Systems, Inc.
P.O. Box 4
Sparta, NJ 07871
Management Sciences

Kranzley & Co.
1010 S. Kings Highway
Cherry Hill, NJ 08034
Banking and Finance

MacNeal-Schwendler, Inc.
7442 N. Figueroa St.
Los Angeles, CA 90041
Engineering-Construction

Management Science America, Inc. (MSA)
Suite 1300
3445 Peachtree Road N.E.
Atlanta, GA 30326
General Accounting
Banking and Finance
Manufacturing
Payroll/Personnel
Tax Tables

Martin Marietta Data Systems
300 E. Joppa Road
Towson, MD 21204
Management Sciences
Manufacturing

McCormack & Dodge Corp.
560 Hillside Ave.
Needham Heights, MA 02194
General Accounting
Banking and Finance
Manufacturing

McDonnell Douglas Automation Co.
P.O. Box 516
St. Louis, MO 63166
Engineering-Construction
Management Sciences

Medical Information Technology, Inc.
255 Bent St.
Cambridge, MA 02141
Electronic Mail
Medical and Health Care

Mini-Computer Business Applications, Inc. (MCBA)
117 S. Brand Blvd.
Glendale, CA 91204
General Accounting
Manufacturing
Sales and Distribution
Tax Preparation

Modular Computer Systems, Inc. (Modcomp)
1650 W. Mcnab Road
Fort Lauderdale, FL 33310
General Accounting
Payroll

NCR Corp.
Bldg. 26
Stewart & Patterson Sts.
Dayton, OH 45479

Nichols & Co.
5839 Green Valley Circle
No. 104
Culver City, CA 90230
Management Sciences

The Office Manager, Inc.
P.O. Box 66596
127 S.W. 156 St.
Seattle, WA 98166
General Accounting
Manufacturing
Sales and Distribution
Word Processing

Personal Software, Inc.
592 Weddell Drive
Sunnyvale, CA 94086
Computer Games
Accounting
Miscellaneous

Plycom Services, Inc.
P.O. Box 160
Plymouth, IN 46563
General Accounting
Payroll

Point 4 Data Corp. (Formerly Educational Data Systems)
1682 Langley Ave.
Irvine, CA 92714
General Accounting
Management Sciences
Text Editing

The Poise Co., Inc.
201 N. Nevada
Roswell, NM 88201
Education

Policy Management Systems
Division of Seibels, Bruce & Co.
P.O. Box 1
Columbia, SC 29202
Banking and Finance
Insurance

Price Waterhouse & Co.
1251 Ave. of the Americas
New York, NY 10020
General Accounting
Banking and Finance

Prime Computer, Inc
40 Walnut St.
Framingham, MA 01701

Pritsker & Associates, Inc.
P.O. Box 2413
W. Lafayette, IN 47906
Management Sciences
Mathematics and Statistics

Program Products, Inc.
95 Chestnut Ridge Road
Montvale, NJ 07645
Auditing

Qpac America, Inc.
One Perimeter Road
Grenier Field
Manchester, NH 03103
Payroll/Personnel

Resource Software International, Inc.
140 Sylvan Ave.
Englewood Cliffs, NJ 07632
General Accounting
Banking and Finance
Education
Management Sciences
Payroll/Personnel

SAB, Inc.
405 Park Ave.
Box 173, FDR Station
New York, NY 10022
Tax Preparation/Planning

St. Joseph's Bank & Trust Co.
202 S. Michigan St.
South Bend, IN 46601
Banking and Finance

Software Assistance
(Formerly Occidental Computer Systems)
P.O. Box 2101
Santa Clara, CA 95051
General Accounting
Medical and Health Care

Software International Corp.
2 Elm Sq.
Andover, MA 01810
General Accounting
Manufacturing
Payroll/Personnel

Sperry Univac
Division of Sperry Rand Corp.

"Companies have finally realized that generalized software can be modified to meet individual needs with relatively little effort."

see total revenues from applications software exceed the revenues from systems software within five to 10 years.

It is generally agreed that the average computer installation in the '80s will spend more money on software than on hardware. Proprietary software, as opposed to software developed in-house, will account for an increasingly large portion of these software expenditures, and applications software will account for more than half of all proprietary software costs before the end of the decade.

Why Go Outside?

We've discussed what we expect will happen in the coming decade. Now let's look at some of the reasons for the recent upsurge in the popularity of proprietary applications software.

There is, of course, the "sameness" factor; companies have finally realized that reinventing the wheel is wasteful and unproductive and that generalized software can usually be modified to meet individual needs with relatively little effort on the part of the user. Let's examine some of the other major reasons for the reversal of thinking about proprietary versus in-house software.

In a recent *Computerworld* article, Martin Goetz, senior vice-president of Applied Data Research, Inc., estimated that in-house development costs are currently running in the range of 10 to 1 over the costs for the same software acquired on the open market. He went on to say that a cost ratio of 100 to 1 is not unusual.

Another significant cost aspect, pointed out in a speech at the 1980 National Computer Conference by Fred Tatar of the Ford Aerospace Corp., is the costs involved in obtaining and supporting qualified personnel. Tatar noted that in his experience, it costs about $15,000 to hire someone to do the job, $20,000 a year to support him and $10,000 if you lose him once he has been trained.

Maintenance Bugaboo

All of these cost-related factors help explain why DP installations are more agreeable to acquiring outside software, but they are not the whole story. A brief look at the management-oriented reports in the *Datapro Applications Software Solutions* service uncovers many other factors that can influence the decision to seek out proprietary software packages.

Maintenance of systems developed in-house is one of the problem areas discussed in this reference service. Several reports address the increasing costs and complexities involved in system maintenance.

Software experts believe that the planning, implementation and maintenance efforts associated with an applications system normally become cost-inefficient after about five years — and even sooner in some cases. Therefore, it becomes increasingly clear that installing a generalized package that has been pretested, is guaranteed and can be maintained at a fixed price may well be more economically feasible than doing the job in-house.

Other historic deterrents to in-house development efforts have been the overrunning of costs and the slippage of delivery dates. These problems do not necessarily reflect poor judgment on the part of management and are often attributable to influences beyond the control of the development group.

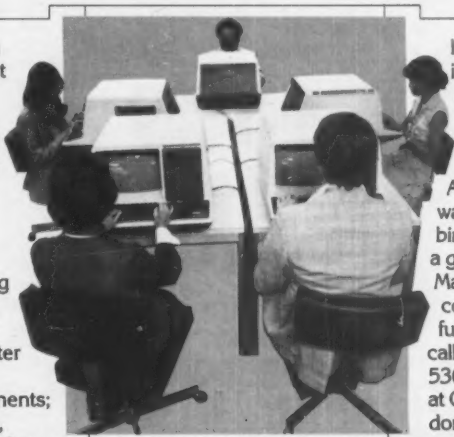
Delays and overruns may result from machine downtime, personnel absences, loss of key personnel, unscheduled shifts of key people to handle emergency situations, expenditures that were not anticipated plus a myriad of other problems. Poor planning is another frequent cause, and much has been written about that problem.

Outside software, on the other hand, comes ready to be installed. The worst that can be expected is that the vendor or user may have to tailor the package to meet the individual user environment and/or to correct some undetected bugs.

We do not want to make light of the software tailoring aspect. With mini/micro software, tailoring is not normally a problem. Because the makeup

Goldmine

for growth-minded software specialists.



When you become a software developer at TRW, you work with computer scientists, engineers, software designers, and other problem-solving types who relish complex and challenging tasks.

They're developing ways to make maxi, mini, and microcomputers work better in distributed data processing environments; ways to design large, dynamic data bases that are also highly reliable; ways to sort, correlate, and display sensor data of all kinds so that military and other decision makers can direct action in real time.

In short, we're a leader in C³I, digital communications, sensor data processing, weapon system software and related disciplines, such as office automation for government applications.

To reduce the cost and risk involved in this state-of-the-art work, TRW has developed a large library of labor-saving software tools. In fact, designing new tools is another activity that may whet your appetite. You may want to help develop architectures, protocols, and operating and applications systems for very large-scale processing networks...or horizontal microcode for special processors. TRW's work-study fellowship in Computer Architecture and Software Engineering combines half-time salary with a generous stipend for a Master's at UCLA— contact Lori Brown. For full-time opportunities, call Bob Chambers (213) 536-3081. Or write them at One Space Park, Redondo Beach, CA 90278.

ADVANCED SOFTWARE TECHNOLOGY

from

A COMPANY CALLED TRW

DEFENSE AND SPACE SYSTEMS GROUP

An equal opportunity employer

of this software tends to be relatively simplistic, it usually is not feasible to modify the functional coding extensively. It would be simpler to look for another package that will do what is required. In view of the wide variety of choices now available, this shouldn't be too hard.

In the case of larger mainframe systems requiring multidivision, multibranch capabilities, by contrast, it is the customer who dictates his needs and the vendor who must satisfy those requirements through software tailoring. In some cases the required modifications are minor and will be performed by the vendor at no cost; in other cases it will be necessary to contract with the vendor to do the work at some fixed cost figure.

All too frequently, the user does not realize the need for extensive modifications until the installation has begun, at which time the user is likely to be trapped because of the initial acquisition agreement. This situation can be avoided through careful planning and thorough investigation of the product being considered for purchase.

The *Datapro 70* reference service carries a feature report called "How to Buy Software Packages," outlining the things you should look for when getting ready to purchase outside packages. Basically, you should seek answers to the following questions:

Do the basic functions of the package meet your needs?

Will the package run on your computer system?

What are the operating system or language requirements?

How well do the detailed capabilities (input, computational capabilities and output) match your requirements?

Is the performance adequate for your needs?

How flexible (portable) is the package?

In what form is the software delivered (is it compatible with your hardware)?

How difficult will it be to install the software?

Will it be easy to use?

Is adequate documentation provided?

What support (if any) will the vendor provide, and at what cost to you?

What is the operational status of the package (has it been fully tested by others)?

What will be the total cost of acquiring and using the package?

What financial arrangements are offered?

The importance of clear, complete documentation could hardly be overemphasized. Because of the end-user aspects of applications software, the operations manual plays an extremely important role in installing and using the software successfully. Many a package has been shelved because the documentation was too difficult to use and the user did not want to take the time to get some help.

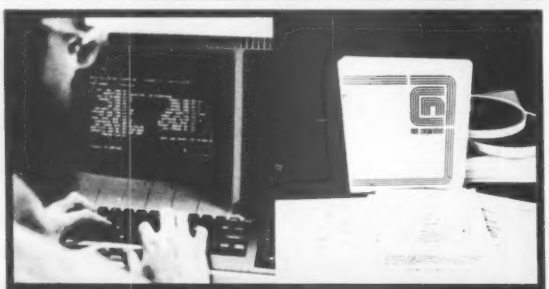
Source Code Key

The availability of source code becomes extremely important when dealing with a company that could fold, since without it the user could be left high and dry in case of a serious problem or if modifications are required. Equally important are adequate training and orientation of the software users. It is up to both the vendor and the customer to make sure that the eventual end users of the software are properly trained and prepared to utilize the software in the manner in which it was designed.

Another significant facet of the task of choosing and acquiring proprietary software is the contract negotiating activity. With mini/micro software, this activity is generally not as significant as in mainframe software acquisition. Because mini/micro software is frequently supplied with the hardware or made available on an "as is" basis, the customer has little recourse and therefore little to gain from an expensively negotiated contract.

Mainframe users, however, should carefully protect themselves when acquiring expensive, sophisticated software systems and/or applications packages. The *Datapro Applications Software Solutions* service contains a feature report that lists the key factors to consider in drawing up a contract for software acquisition:

- The contract should specify the guarantee period after installation (and/or acceptance) during which the supplier will correct all bugs and furnish all required maintenance at no additional cost.
- If the package is found unacceptable, the contract should permit its return within a specified period after installation at no cost to the user.
- Any penalty for late or incomplete



ON-LINE MANAGEMENT INFORMATION SYSTEMS

MS-11 Manufacturing System

- Inventory Control
- Purchasing
- Bills of Material
- Work Order Status
- Manufacturing Cost
- Routing/Capacity Planning
- Requirements Planning

FS-11 Financial System

- Order Entry/Invoicing
- Sales Analysis
- Accounts Receivable
- General Ledger
- Accounts Payable

NCA's MS-11 and FS-11 Systems are fully integrated, on-line information systems designed for the Digital Equipment Corporation PDP-11 computer family. The systems provide data security, flexible on-line inquiry and report program generation.

For more information, write or call NCA, the Software Specialists.

nca corporation

388 Oakmead Parkway
Sunnyvale, CA 94086 (408) 245-7990

P.O. Box 500
Blue Bell, PA 19424

SPSS, Inc.
444 N. Michigan Ave.
Chicago, IL 60611
Mathematics and Statistics

Stockholder Systems, Inc.
1965 N. Park Place
Atlanta, GA 30339
General Accounting
Banking and Finance

Systems Research, Inc.
2400 Science Parkway
Okemos, MI 48864
General Accounting
Education
Payroll/Personnel

Systonetics, Inc.
Suite 70
600 N. Euclid St.
Anaheim, CA 92801
Management Sciences

Technical Analysis Corp.
120 W. Wieuca Road N.E.
Atlanta, GA 30342
Payroll/Personnel
Sales and Distribution
Word Processing

Tektronix, Inc.
P.O. Box 500
Beaverton, OR 97077
Graphics Design

Timberline Systems
10550 S.W. Allen Blvd.
Beaverton, OR 97005
Specialized Accounting Systems

Time Share Corp.
630 Oakwood Ave.
West Hartford, CT 06110
Education

Tymeshare Banking Systems
11 W. Oxmoor Road
Birmingham, AL 35209
General Accounting
Banking and Finance

UCLA Health Sciences Computing Facility
University of California
Los Angeles, CA 90024
Mathematics and Statistics

United Computing Systems, Inc. Business Information Products
2525 Washington
Kansas City, MO 64108
Banking and Finance
Tax Planning

University Computing Co.
UCC Tower, Exchange Place
Dallas, TX 75235
Banking and Finance
Manufacturing

Versatek, Inc.
2805 Bowers Ave.
Santa Clara, CA 95051
Plotting Graphics

Wang Laboratories, Inc.
One Industrial Ave.
Lowell, MA 01851

Warrex Computer Corp.
1780 Jay Ell Drive
Richardson, TX 75081
General Accounting
Banking and Finance
Payroll/Personnel
Sales and Distribution

Warrington Associates
5600 Lincoln Drive
Minneapolis, MN 55436
Banking and Finance

Weiland Computer Group, Inc.
Suite 950
1515 W. 22 St.
Oakbrook, IL 60521
Banking and Finance

Western Electric
P.O. Box 25000
Greensboro, NC 27420
Electrical Engineering
Graphics Design
Mathematics and Statistics

Bob White Computing & Software, Inc.
830 Diane Lane
Naperville, IL 60540
Banking and Finance

Wood & Tower, Inc.
P.O. Box 28
Research Park
Princeton, NJ 08540
Construction

"In acquiring sophisticated software, it is wise to seek the assistance of a lawyer familiar with software contract negotiations."

delivery and installation should be spelled out.

• A clear description of any reduced rates or extra support should be included if your installation happens to be a product test site for the vendor.

• All agreed-upon modifications to the package, together with the responsibility and deadlines for making them, should be specified in detail.

• Objectional restrictions on the modification or use of the package should be eliminated from the contract.

• Installation support, training and maintenance promised by the vendor should be spelled out in detail in the contract, together with the additional costs of the services, if any.

• The names of specific individuals who will supply the on-site support, or the specific qualifications these individuals will have, should be specified in the contract. (It is particularly desirable in the case of a new, untried package to have the people who developed it present at installation time.)

• The method to be used for correcting bugs discovered after installation, and the guaranteed response time for correcting them, should be specified.

• The contract should specify exactly what items will be delivered in what forms (e.g., source decks and listings for all programs plus all relevant documentation at installation time, followed by modified or improved versions of the package whenever these are made available to new buyers).

• The buyer should be willing to agree to reasonable clauses that protect the supplier's proprietary rights by forbidding resale or other unauthorized distribution or use of the software.

• Payment for the package should be contingent upon delivery of all the promised products and services and successful operation of the package in your installation.

Probably no software package vendor will be willing to make all these concessions; he's in business to make a profit, too. But by playing the role of a tough, sophisticated negotiator and demanding everything you can get, you're likely to save more money and get far better support than the buyer who meekly signs the standard contract form. In the acquisition of sophisticated software systems, it is wise to seek the assistance of a lawyer familiar with software contract negotiations.

The Wise Software Buyer

The prospective buyer of an applications software package should take a long, hard look at what is available these days. More than 5,000 proprietary software packages are currently on the market. Of these, more than 3,000 are applications-oriented in nature. Over 50% of the current packages are geared for use with minicomputers and microcomputers, and this percentage is growing rapidly.

The wise buyer will shop around carefully before committing himself to any one package. In summary, he should:

(1) Determine his requirements.

(2) Gather information about all the available packages that appear to meet his requirements.

(3) Narrow the field by rejecting unsuitable packages.

(4) Perform a detailed evaluation and comparison of the remaining packages.

(5) Talk to other users.

(6) Conduct benchmark tests (or use each package for a trial period).

(7) Select the most suitable package.

(8) Negotiate a sound contract.

(9) Install the package.

(10) Check the results.

By following these guidelines and keeping aware of what is available in the applications software marketplace, the software user of the 1980s can face the approaching decade confidently, secure in the knowledge that he can stay on top of all the dynamic changes that will be taking place. The next five years will be the most critical, and a firm footing now, with a planned and plotted course for the future, is likely to mean the difference between organization and chaos.□

ARIZONA OPPORTUNITIES

New Positions . . . IBM Environment

Top computer professionals for a top software company. That's what we're looking for at Capex. Capex is a leading worldwide developer and marketer of proprietary computer software products. Rapid growth has created the opportunities for ambitious people to participate in the development of major new software products.

TECHNICAL REPRESENTATIVE

As a TECH REP, you will be the initial telephone contact providing technical assistance to our customers worldwide. Positions require good analytical skills, excellent communication ability and dedication to end-user support. Requires background in programming, systems, or operations. TSO, CICS, or IMS experience a plus.

TROUBLE SHOOTERS/QA

You'll handle the testing or troubleshooting of one or more system-oriented products installed on a wide variety of systems. Knowledge of any of the following would be helpful: DOS/VS & VSE, OS/VS, VS1 and MVS. Positions require good diagnostic or analytical skills in BAL or COBOL. Exposure to TSO, CICS, IMS, or systems programming desirable.

DOS SYSTEMS PROGRAMMER

We're looking for an aggressive Systems Programmer to be a key part of a highly talented team in designing and developing major technical programs. Recent DOS systems programming experience or strong assembler background needed. Also required is a minimum of three years DOS/VS programming experience.

SOFTWARE SYSTEMS SPECIALISTS

- Systems Programmer - Play a key role in developing sophisticated new products for IBM OS/VS environment. Two years ALC experience required. Design background and a familiarity with DASD protocol will be helpful.
- Senior Programmer - System Products We offer excellent career potential for an individual with solid assembler skills. Position provides substantial growth in technical areas of OS/VS, MVS, IMS, and CICS.
- Senior Programmer - Compiler Products Take a leading technical role in enhancing and developing products for our OPTIMIZER product line. Requires strong IBM ALC and COBOL programming skills. Prefer Systems Programming experience in COBOL compiler and/or CICS internals areas.

APPLICATIONS PROGRAMMER

Work as a key part of our software team developing new products for the IBM OS/VS environment. Minimum one year programming experience, ALC background preferred. COBOL acceptable.

We offer rewarding challenge, competitive salaries and an excellent benefits package. If you think you qualify for one of these Phoenix-based positions, send your resume with salary requirements to:

HUMAN RESOURCE PLANNING
CAPEX CORPORATION
4125 North 14th Street, Phoenix, Arizona 85014

Offices located in major cities worldwide.

Equal Opportunity Employer Male/Female/Handicapped

By Ronald G. Ross

DATA MANAGEMENT FOR THE 80s

Experience with data dictionaries over the past several years has demonstrated that it is considerably easier to understand the functions of a data dictionary system than to precisely define what the data dictionary is. There seems to be something unsettling about definitions that contain phrases like "data about data" or "a repository of definitions for data" or "meta data for the data base." Such descriptions seem either to say too much or to say nothing at all.

For this discussion, a data dictionary system is viewed simply as a tool for

PART I
Data Dictionaries: Basic Concepts

managing corporate data resources. A data dictionary contains that information needed by this system to perform its various objectives.

This approach places the emphasis squarely on function, rather than on methodology or dictionary structure. An implication is that in selecting a data dictionary, it is more important to understand *what* a dictionary does rather than *how* it goes about doing it.

This sharply contrasts with evaluation strategies for selecting software systems of other types. In data base management system (DBMS) selection, for example, it is neccesary to examine how a package solves problems of a general type in order to determine whether the system can solve the particular needs of the given corporation.

The reason for this is that a DBMS is a generalized software tool which must be designed and tailored to each particular application problem. The number of these potential applications is practically unlimited. For each given case, the internal mechanisms of the DBMS may make a great deal of difference.

Some applications, for example, are best served by inversions of record values, whereas others are better served by interrecord pointers. The purchaser must decide which is the best for his applications.

A data dictionary system is not generalized in this sense. In fact, it is designed to solve one rather specific type of "application" problem, the area of data resource management. Its internal mechanisms make little difference, since the user is not asked to tailor the system to this application at a detailed level. The decision to acquire a data dictionary system is essentially the same as purchasing a turnkey system — the emphasis is on *function*.

This does not imply that the problem area is small; on the contrary, it has proven extensive. Nor does it imply that it is easy to define precisely the needs of this application. Testimony to this fact is the variety of commercial packages now on the market.

Despite this, there is in fact a common theme to all the offerings, which reveals itself most clearly in similarity of function. One might say this collection of functions delimits the traditional approach to the data dictionary problem.

Will the traditional approach to the data dictionary problem be eventually supplanted by new data dictionary technology? Recent evidence suggests that the answer to this question is undoubtedly "yes." There are, in fact, both theoretical and practical grounds to suspect that the data dictionary problem is not a specific "application" problem after all, but instead a generalized problem, in essence equivalent to the DBMS case. Although this does not invalidate the more traditional functions examined here, it certainly adds to them new and exciting dimensions.

Inseparable Problems?

During the first decade of data base development, DBMS were often viewed as more or less the final solution for all data management ills. During the latter years of the 1970s, however, data dictionary systems began appearing in increasing numbers espousing DBMS-like themes.

For example, data dictionary systems purported to assist in "the management of data resources," something to which DBMS claimed all along. Many professionals thus began asking: If data base, why also data dictionary?

Even more recently, data dictionary systems have begun encroaching on non-DBMS territory — specifically espousing potential for those organizations with no data base development at all. Now professionals have begun having to ask a second question: If *not* data base, then why data dictionary?

Surprisingly, from an organizational point of view, the answers to these two questions are largely equivalent. In general, a data dictionary system is aimed at improving DP productivity by optimizing control and use of the definitions on which systems are built. Although more tangible in the case of DBMS (schemas and subschemas are explicit types of definitions), all systems, data base or not, are nonetheless built on definitions.

By automating the management of definitions, a number of documentation needs are served. These needs arise in the enhancement and maintenance of large application systems, increasingly a difficult and costly concern within every DP installation, whether data base-oriented or not.

Furthermore, once the management of definitions is automated, a springboard is established for offering other types of system-related services. These services, and the occasions for their use, are largely indentical for both data base and nondata base applications. The enforcement of standards and the generation of programmatically usable data descriptions are prime examples.

Back to Basic Questions

To return to the basic questions about data base and data dictionary, the first really asks how DBMS and data dictionary systems differ. A close examination of their respective functions, classed within a single breakdown of data management goals, is shown by the analysis in Figure 1. This analysis shows that the real distinctions are actually ones of level, rather than ones of purpose.

What about data dictionaries for the non-DBMS installation? Traditionally, the bulk of data dictionary thinking has focused on the DBMS-supported organization. There are two reasons for this. First, data base evolution, in and of its own momentum, naturally suggested additional areas in which more sophisticated data management techniques were possible. Data dictionary systems have arisen largely in response to these DBMS-oriented ideas.

Secondly, DBMS in their own right make possible more centralized management of data resources. They thus provide a solid foundation for the elaboration of basic data management capabilities.

In non-DBMS organizations, there is less centralization of control and consequently less on which a centralized tool can readily operate. Similarly, since system management is generally more diffuse, specific needs are ironically often more difficult to pinpoint.

As a result, the need for a data dictionary system has been more slowly perceived in non-DBMS organizations than for those with DBMS. Acceptance by the non-DBMS community has therefore been slower.

Picture Changing Fast

This picture is changing rapidly. Many non-DBMS users — whether they avoided data base because of smaller machine size or because they are interested in supporting only existing application software, or because they are suspicious of how existing DBMS fit into an evolutionary mainstream — are looking increasingly toward data dictionary to solve basic system management problems.

Even DBMS users, many of whom have been surprised to find what a large percentage of their system resources persist under nondata base schemes, are using dictionaries more and more for all systems, not just data base alone.

Perhaps the most profound concept that has emerged from recent data dictionary experience is that the dictionary represents the most viable of all alternate routes for easing the pains of migration to data base technology by non-DBMS organizations. This is *not* because a data dictionary can help lessen the conversion impact of DBMS (or more accurately, the reimplementation impact), but rather because a dictionary can help set in motion a centralization of administration and control — the most difficult to achieve in practice of all data base requirements.

In other words, data dictionaries help create an organizational climate for data base by encouraging the formation of a data administration team and by planting the seeds of the data resource concept, the essential first step in successful data base projects.

These are not just idle claims; these are the findings of pioneers that have shown the way for successful data dictionary and data base projects during the 1970s.

The Problem

Every class or type or software is specifically designed to address one or several identifiable problem areas. DBMS software, for example, focuses heavily on the separation of logical and physical data base views, allowing greater protection for the integrity of the data base as well as improvement in DP productivity. A second focus is the elimination of data redundancy, which is achieved by complex and generalized data structures.

For data dictionary systems, a central theme is improving the authenticity of information, or documentation, about a corporation's application systems. A dictionary does this by bringing an end to the unnatural isolation that has always characterized DP documentation strategies.

The typical DP installation keeps its documentation within a library. Although often housed within the same building as the computer, this physical proximity is usually the full extent of any direct relationship or integration with the actual software systems it

---

Ronald G. Ross is editor of "Data Base Newsletter," a bimonthly publication of Performance Development Corp., Princeton, N.J.

This article is excerpted with permission from Data Dictionaries and Data Administration: Concepts and Practices for Data Resource Management *by Ronald G. Ross, © 1981 by AMACOM, a division of the American Management Associations. All rights reserved. This book will be available from the publisher next spring.*

Data Resource Management Goal	DBMS Function	Data Dictionary Function
To build and control corporate data resources.	Management of corporate data resources.	Management of the definitions and documentation for corporate data resources.
To organize models of corporate information.	Physical support for relationships between data that is meaningful to the corporation.	Support for documenting the types of information within corporate data resources, as well as the system components that support them.
To improve the accessibility of corporate information.	Physical support for data access, by keys and interrecord relationships.	Support for investigating the types of corporate information available, as well as the means by which they are supported.
To increase the longevity of system and data resources, to facilitate their maintenance and enhancement and thus to protect the corporation's investment in them.	Separation of logical and physical data base views, making application programs independent of changes in data organization and making new applications easier to implement.	Provision of tools to accurately assess the specific impact of data and system changes and to determine what the content and extent of existing resources are.
To improve the accuracy and consistency of corporate information.	Elimination of data redundancy through systems-supported methods of data integration.	Elimination of the artificial segregation between system documentation and system driving definitions.
To bring coordination and consistency to information system implementation efforts.	Centralization of data definition.	Centralization of definition management, standards enforcement, naming procedures and modification control.
To improve the productivity of the data resources staff.	Provision of programming tools to implement complex application problems, while isolating systems from the impact of change.	Provision of programming aids, such as ready made data definition and trustworthy impact assessment tools.
To protect the data resource.	Support for access security and data base integrity, back-up and recovery.	Support for protection of all definitions on which information systems are built.

FIGURE 1. A COMPARISON OF DBMS AND DATA DICTIONARY FUNCTIONS, WITH A COMMON CLASSIFICATION OF DATA RESOURCE MANAGEMENT GOALS

purports to describe. Documentation management is usually 100% manual, and all update actions are totally distinct from the actual changes occurring in the described systems.

Furthermore, in system development, the documentation task typically is the last to be undertaken. Standards and guidelines for this documentation usually exist, but whether they are followed is another question entirely.

Because of time constraints, disinterest or lack of coordination, the resulting documentation is often uneven, incomplete and disorganized. Even where standards are met, the resulting material may be too voluminous for easy use, too disorganized for ready reference or too scanty for comprehensive analysis.

An additional problem is that there is no assurance that the documentation is up-to-date at any given point in time. Keeping documentation current depends entirely on the development or maintenance personnel remembering (or being reminded) to go to the documentation library and make the necessary updates. There is nothing automatic or inevitable about this taking place.

Even if the documentation library is kept completely current with the software it describes, there is no guarantee that the information it contains will be easy to use. Answers to questions about a corporation's system resources are useless unless they can be readily formulated as the need to know arises.

This is particularly true in system design and development. Buried within existing documentation is often a wealth of information that might be used in planning how new data needs might be met. Unfortunately, existing documentation strategies are usually static and cannot satisfy more dynamic needs of this sort.

One final problem area exists with traditional documentation practices, which is particularly relevant for users migrating to a data base environment. At any given installation, documentation standards have evolved over many years, during which all system development was nondata base. Such applications, and thus their documentation, tend to be program-oriented and relatively sparse on data definition control.

In data base systems, the emphasis is necessarily reversed — *data* becomes the focal point of the system, and all other system components, including programs, revolve about it. This change of perspective is often hindered by traditional practices, rather than helped.

Problems with traditional documentation strategies can be summarized as follows:

- Management of documentation is a largely manual process, which is separated from events or changes occurring within the source or object code of the application systems it describes.
- There is no automatic guarantee that existing documentation is up-to-date, synchronized with executing system configurations or accurate in representing the data and system resources of the corporation.
- Managing and motivating the pursuit of comprehensive documentation is a difficult and time-consuming affair.
- For a number of reasons, the resulting material may be difficult to use or ineffective in meeting the dynamic needs of the corporation. Real questions arise about the authenticity of the information it contains.

How does a data dictionary system address these problem areas? The key ingredient in the data dictionary approach is the merging of those definitions used to drive actual system software with that material used to produce system documentation.

For example, if the same file description used within the object versions of programs executing in the machine is also used to produce a listing of file contents, then it can be guaranteed that this listing is accurate, up-to-date and completely in sync with the actual system. It cannot help but be totally authentic.

There are problems, of course, in using the executing object version of a file description to produce documentation, but this is an ideal that can be successfully approximated with various dictionary techniques. Distinctions can be drawn, in fact, between the degree to which various types of dictionary systems can achieve this goal, as Part II will show.

Creating the Basic Dictionary

The basic level at which a file description (FD) can be incorporated into the dictionary is by holding the source definition. Whenever a program is compiled, this source definition is extracted from the dictionary, either directly or indirectly, and included in the compilation.

Within the dictionary system itself, whenever a query about the content of the file arises, this same FD can be used to produce the necessary information.

Even more sophisticated capabilities arise where the dictionary is able to keep track of programs using the FD and, by implication, the associated file. By ascertaining which FDs a given program is using during compilation, for subsequent reporting the dictionary system can invert the information in order to show all programs by which a given file is used. This type of usage reporting is often referred to as *cross-reference* or *where-used* mapping.

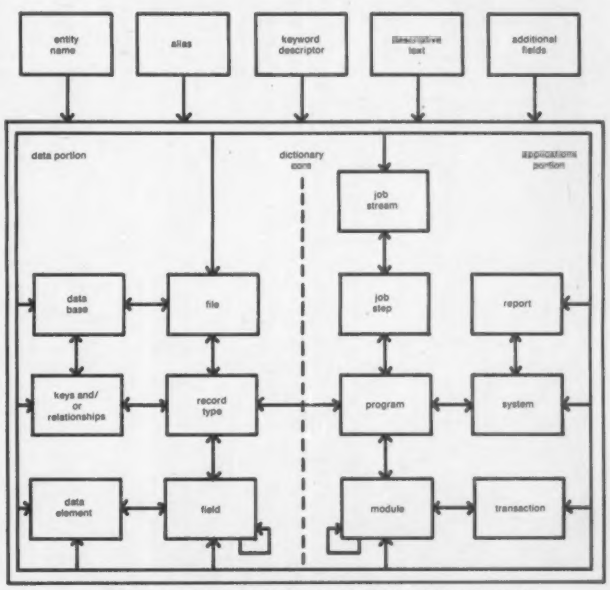


FIGURE 2. A SCHEMATIC DIAGRAM FOR THE BASIC DATA DICTIONARY DATA BASE

To take this a step farther, since files themselves are composed of many fields, or data elements, then by cross-referencing an additional level (i.e., program to file to *data element*) information can be produced about all the programs by which a given data element is used. This represents two levels of where-used reporting.

How does the data dictionary system learn what files a given program uses? In nondata base systems, this is problematic. The information is certainly known at compile time, but the compi-

"The data dictionary is a highly interwoven structure with many components and relationships."

lation process generally occurs outside the realm of the data dictionary system. It can be gathered by inspection of program source code, but it is not clear how and when this inspection should be triggered. Execution-time monitoring, of course, is impractical.

In short, since there is no clear-cut point at which the dictionary can automatically learn about file usage, extraordinary methods are necessary. These methods are generally procedural in nature and involve rigorous control by the data administrator.

More Straightforward

In a data base environment, where control over data and system resources is less diffuse, the solution is more straightforward. Under a DBMS, programs do not typically make their own file or record definitions, but are instead matched with a *subschema* specification in which the definitions are made for them. Subschemas are common to the majority of DBMS, though they often parade under different labels.

The subschema declaration is an obvious source of information for the data dictionary. Not only does the subschema specify which files or record types the program may access, but often the subset of fields, or data elements, to which it is permitted access as well. This is a ready-made cross-reference mapping.

Furthermore, since the subschema is enforced by the DBMS during execution, there is no question about whether the mapping is accurate. The dictionary system is provided with a ready means of satisfying the program-to-file-to-data element mapping requirement.

Using the Basic Dictionary

Why is cross-reference information of the type described above useful? Suppose the length or "picture" of a particular data item needs to be changed. This implies that all the programs using that item must be modified, or at least recompiled. The ability to accurately determine on a momentary and automatic basis all the application software so impacted is an invaluable asset to maintenance.

A similar use is in resolving problems in data integrity. Suppose, for example, that an unknown errant program fails to conform to standards in creating or maintaining a field in a certain file. A where-used listing of all accessing programs can greatly streamline the research and correction effort.

So far in this discussion, the only type of cross-reference mentioned has been the program-to-file-to-data element mapping. A typical data dictionary has many such relationships, each serving in its own fashion to provide information needed in management of the corporation's data and system resources.

Other examples of relationships within a typical data dictionary include the following:

- System-to-program-to-module.
- User-to-report-to-data element.
- Logical terminal-to-transaction-to module.

There are many others. These examples suggest that the data dictionary is in fact a highly interwoven structure, with many components and relationships. The management of these structures, as performed by the data dictionary system, is, in fact, a complex undertaking.

Information Sources

Producing information for the dictionary about these components and relationships is not always a straightforward task. Although FDs and subschemas come ready-made for this need, information of other types is not so readily available. Often, the dictionary user is required to complete special forms, which are then processed by special dictionary input subsystems. Dictionary involvement in these areas is only minimally automatic.

In a data base environment, however, one additional source of ready-made information exists: the schema or data base definition.

The schema contains extensive information about the contents of the organization's data base, as well as

DOS/VSE and FBA Support Available Now

Tower Systems' TFAST Dataset Management System gives DOS/VSE users state-of-the-art features no competing system offers. Check what YOU have compared to the TFAST features listed below.

- ☐ Common catalog for tape/disk datasets.
- ☐ No re-naming or replacing of IBM phases.
- ☐ SVA Resident—eliminates library fetches for improved throughput.
- ☐ All operator communication handled in the problem program . . . eliminates LTA convention for improved throughput.
- ☐ FBA DASD support.
- ☐ Automatic File Re-blocking with no changes to user programs for increased throughput.
- ☐ Three levels of password protection to control access to datasets and to the catalog.
- ☐ On-line real-time access AND UPDATE facilities via CICS.
- ☐ Completely integrated vault-management system . . . automatically controls and reports on the movement of datasets to off-site storage locations.
- ☐ Complete Audit-Trail reporting system of all dataset activity. Plus DOS/VSE Job-Accounting reports.
- ☐ Auto-catalog eliminates manual cataloging of dataset-names.
- ☐ Remote-Console-Support . . . all TFAST messages and replies may be routed to a CICS terminal.
- ☐ Dataset Integrity . . . TFAST/VSE guarantees correct version of dataset will be used as input.
- ☐ Automatic-Volume-Recognition eliminates need to assign physical devices to logical sys-no, or to dedicate drives to specific partitions.
- ☐ Autostop puts to sleep and wakes up partition if dataset required is not available . . . frees console . . . user defines intervals.
- ☐ Early-Device-Release . . . frees tape drives at close time, making them available to other partitions.
- ☐ Dataset-Protection, the system will never permit an active version of a dataset to be written on . . . tape or disk.
- ☐ Mixed-Media Support. TFAST allows versions of datasets to be tape, disk or a combination of both.
- ☐ No JCL or operating system changes.
- ☐ Automatic Catalog Recovery . . . TFAST will re-build your catalog if it is inadvertently lost.
- ☐ Assgn and extent JCL eliminated if desired.
- ☐ Complete 17-character file names.
- ☐ Automatic Secondary Allocation.

TOWER SYSTEMS, INC. 19782 MacArthur Boulevard, Suite 365, Irvine, CA 92715 ▪ TWX: 910-595-2780 ▪ Phone: 714/752-5263
SALES OFFICES: New York City, (212) 724-0620 ▪ Chicago, (312) 255-9560 ▪ Dallas, (214) 387-3534 ▪ Atlanta, (404) 767-8290

- ☐ Send more information about TFAST Dataset Management.
- ☐ Tell me about your trade-up plan from obsolete tape-disk management systems.
- ☐ Have representative contact me.

Name_____ Title_____
Company_____
Address_____
City_____ State_____ Zip_____
Telephone ( )_____ CPU_____ Op. Sys._____

"Dictionary terminology, unless carefully chosen, can prove unduly restrictive in accomplishing larger objectives."

about its structure. There are innumerable uses to which this information can be put, particularly for more sophisticated types of programming and access support. Besides forming a basic structural core for the dictionary itself, incorporation of the schema into the data dictionary represents one of the most fertile grounds for data dictionary evolution.

What of nondata base applications, where central and independent definition is lacking? Whereas the control potential is diminished, the need to know about the content, status and usage of data and system resources is not. The assistance the dictionary can provide in these areas is indeed valuable.

There are many additional services the dictionary system can offer, far beyond the original documentation problem discussed above. Many of these services arise from the same fundamental theme, the incorporation of data and system definitions directly into the dictionary.

The Dictionary Data Base

In the discussion above, some of the components of a typical data dictionary were mentioned — data elements, files, record types and programs, for example. These components have natural relationships with one another arising from the manner in which DP systems are themselves constructed. For example, the system components (data element, file and program) are related in certain definable ways because programs use files, which are composed of data elements.

The information about all these components and all of their relationships make up the data dictionary data base. This data base, which may or may not be managed by a DBMS per se, has an established structure, or *data model*, under which information about the various components is stored.

Unfortunately, descriptions of this data base are often confusing. Consider, for example, that there is a data dictionary record type which contains information about application system record types — a *record type record type*. This indicates that considerable care must be taken in the choice of terms used to describe the dictionary and its structure.

Furthermore, a dictionary is often used to collect information about diverse objects within the overall resources of the corporation, not just about application systems. Dictionary terminology, unless carefully chosen, can prove unduly restrictive in accommodating these larger objectives.

For this reason, the relatively nondescript term *entity* is often chosen to denote those things about which information in the dictionary is kept. In general data base usage, an entity is simply a person, place or thing. Employees, departments, skills, orders and invoices are entities of different types.

In the dictionary data base, of course, basic entities include the various system components: files, data elements, record types, data bases and programs. All data dictionary systems support automatic collection of information about these basic entity types. Many dictionary systems also support automatic collection of information about additional entity types: modules, systems, transactions, reports, screens, forms, users, job streams and job steps.

Should the dictionary user find these entity types wanting, some dictionary systems allow definition of additional entity types, of whatever form the user desires. For example, to assist in the development of application projects, a dictionary user might define entity types for project stages, target users, operational procedures, employee time sheets, usage documentation and so forth. Once these entity types are defined to the system, automatic dictionary facilities for storage and reporting may be used to maintain and access the information.

The dictionary system may also allow the user to define relationships for these entity types, either with entity types that are basic to the system or with others that the user himself has defined. To manage the system devel-

ALMOST PERFEC.

The MAGIC WAND™ is the most powerful, most flexible, most reliable, *most usable* word processing software available for a CP/M®-based computer.

That's not bragging. That's just telling it like it is.

The MAGIC WAND *is* the best word processing software ever written for a microcomputer. It can do more work in less time with higher quality than any other product you can buy.

The MAGIC WAND is a rock solid piece of software. The command structure is simple and logical and complete. We have not tossed in features without thought to the overall design of the package. Nor have we included any feature that is not thoroughly implemented. The programs are crash-proof and completely reliable.

And the system is supported by what we are told is the best user's manual ever produced for microcomputer software. It contains a step-by-step instructional program designed for the novice. The trainee uses sample files from the system disk and compares his work to simulated screens and printouts in the manual.

Support doesn't stop when you buy the package. As a registered user, you receive our bi-monthly newsletter which answers questions, reports upgrades and teaches new applications of the MAGIC WAND.

It's through a lot of hard work that we are able to offer you a product that is "almost perfect," but we aren't about to stop working until we can say that the MAGIC WAND *is* perfect.

Full screen text editing

The MAGIC WAND has probably the most responsive and easy-to-use editor available for either a serial or DMA terminal. It uses only single stroke control keys to give command and takes advantage of the special function keys on your terminal whenever possible. In addition, you can set up library files with coded sections that you can merge by section name.

Full text formatting commands

The MAGIC WAND allows you to set the left, right, top and bottom margins, page length, indentation, paragraph indentation, (incuding "hanging" paragraphs), text left flush, right flush, justified (two ways), literal or centered, variable line and pitch settings, variable spacing (including half lines), bold face, underlining (solid or broken), conditional hyphenation, sub- and superscripting. You may change any of these commands at run-time *without reformatting the file.*

Merging with external data files

You may access any external data file, with either fixed length or sequential records. The MAGIC WAND converts the record into variables that you define and can use like any other variable. Of course, you may use the data for automatic form letter generation. But you can also use it for report generation.

Variables

You may define up to 128 variables with names of up to seven characters. The current value of a variable may be up to 55 characters, and you may print it at any point in the text without affecting the current format. Although the MAGIC WAND stores the variables as strings, you may also treat them as integer numbers or format them with commas and a decimal point. You may increment or decrement numeric variables or use them in formatting commands.

Conditional commands

You may give any print command based on a run-time test of a pre-defined condition. The conditional test uses a straightforward IF statement, which allows you to test any logical condition of a variable. You may skip over unneeded portions of the file, select specific records to print, store more than one document in a single file, etc.

True proportional printing

The MAGIC WAND supports proportional print elements on NEC, Diablo and Qume printers. Other formatting commands, including justified columns, boldface, underline, etc., are fully functional while using proportional logic.

Available on 8" soft-sectored and 5 1/4" Northstar or Micropolis (hard or soft sectored) diskettes, as well as ONYX hard disk. Terminals supported include—ADDS, Beehive, Cromemco, Dynabyte, Hazeltine, Heath, Imsai, Intertec, Lear Siegler, Microterm Act V, Perkin Elmer, Sol VDM1, Soroc, TEC, TEI, Televideo, TRS80 Mod II, Vector Graphics, plus a variety of video boards.

small business applications, inc.

3220 Louisiana • Suite 205 • Houston, Texas 77006 • 713-528-5158

CP/M is a registered trademark of Digital Research Corp

opment project, for instance, the user might define a relationship between the entity types "project stage" and "employee time sheet." In so doing, a basic structure for a project control system is created.

This suggests that the data dictionary system may allow the high-level definition and management of special-purpose data bases. This relatively new development in data dictionary technology, called *extensibility*, is one with significant promise for the future. It is also one that implies data dictionaries are becoming "generalized" software tools, rather than the "application"-specific tools they have been in the past.

The Basic Dictionary Structure

Figure 2 illustrates the basic structure, or data model, that might be found in the typical data dictionary. This structure is composed of a number of system-related entity types, as well as many relationships between them. This is not the only structure that one might choose for modeling the architecture of DP systems; even among commercial dictionary systems a variety of such structures exist.

The dictionary data model is surprisingly complex. This is, in fact, even a simplified version of what one might find in the typical dictionary. From a systems perspective, coordination of information for the various entity types, as well as for the relationships among them, is no trivial task. What this really suggests is that the data dictionary is actually a high-order data base problem in its own right.

Fortunately, for dictionary selection, it is not necessary to understand every implication of dictionary data base management in order to make an intelligent choice between packages. This highlights the earlier point that the emphasis in data dictionary selection rides on function rather than on internal mechanisms. For those considering their own dictionary system development, this structural complexity is reason enough to take pause.

In actual operation, the dictionary structure becomes a populated dictionary data base, as illustrated in Figure 3. In the segment of the dictionary shown, four of the 13 core dictionary entity types appearing in Figure 2 are represented by actual entity *occurrences*. A where-used report, for the field JOB TITLE, tracing various relationships between these entity occurrences, is also shown.

Figure 3 helps illustrate the critical distinction between entity *occurrences*, or information about actual system components, and entity *type*, the classification for this information. The structural diagram in Figure 2 was based solely on type; the populated dictionary in Figure 3 shows actual occurrences of these types. The term *entity* (or entities) by itself insufficiently makes this sometimes troublesome distinction and should be avoided.

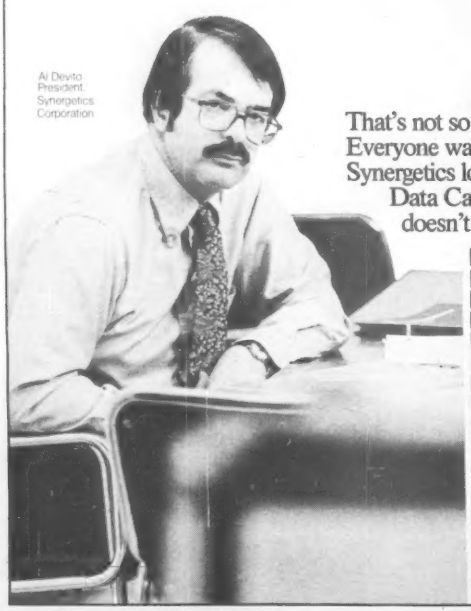
PART II
Merging the Data Dictionary With Other Processes Of Data Management

As discussed in Part I, a data dictionary system (DDS) improves upon traditional documentation strategies because it facilitates the use of the same definitions that drive production systems for the creation of actual documentation products. The quality and reliability of the documentation, as well as the precision of control that can be exercised over the data resource, are thereby greatly enhanced.

Not surprisingly, this consolidation of definitional and documentation strategies can be pursued to any of several increasingly more complete configurations. Important and unexpected capabilities arise as the consolidation tends toward the more comprehensive.

In the following discussion, three configurations of the overall data management complex, distinguished by the degree of consolidation of dictionary processes with the remainder of the software complex, are identified. These configurations respectively identify *passive*, *active* and *in-line* dic-

"Practical people want a high rate of return on their software investment."



Al Devito
President
Synergetics
Corporation

That's not so hard to understand.
Everyone wants value for their money. That's how Synergetics looks at it and that's why we designed Data Catalogue 2, the Data Dictionary that doesn't need a DBMS.

CWE/

☐ I agree, send more information.
☐ Call me.

Name Phone No.

Company Computer

Street

City State Zip

Synergetics

One DeAngelo Drive, Bedford, MA 01730 (617) 275-0250

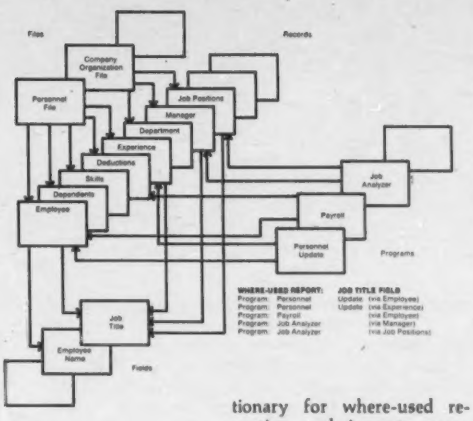


FIGURE 3. A PARTIALLY POPULATED DICTIONARY DATA BASE, WITH WHERE-USED REPORTING

tionary roles.

Whereas the first of these represents the traditional, and certainly prevalent role, powerful evolutionary forces are pushing the DDS more and more strongly toward the other two.

The Passive Dictionary Role

A *passive* data dictionary system is one that is basically unaware of other systems and processes that operate within the same overall machine environment. In other words, the DDS has no direct interaction or involvement with langauge compilers, the DBMS or the operating system of the machine.

The dictionary thus conceived acts in effect as a passive repository of data and system definitions, in or near source form, which are offered more or less openly to "external" needs. The appropriateness and consistency of their use in meeting these external needs are mostly determined by decisions made outside its own operational sphere.

Significant characteristics of the passive data dictionary include the following:

- Even though the dictionary is used to "hold" data base schemas and is able to produce documentation products from them, it does not ensure that these source data base definitions are always those that are compiled to produce the object defintions controlling the execution-time processes of the DBMS. Nor does any reciprocal exchange take place — the dictionary version of the schema is *not* consulted during the compilation process to ensure the continuing consistency of the definitions.

In other words, it is entirely possible to modify and compile a schema whose origin may or may not be from the dictionary without the dictionary system ever realizing that the event has taken place.

- Similarly, even though the dictionary may hold subschema definitions, it is unaware of subschema processing events. Thus the dictionary takes no overt action to ensure that its subschema definitions are actually those compiled to produce object versions for the DBMS, nor does the DBMS subschema compiler cross-verify the subschema it processes to ensure its continuing consistency with the "official" dictionary subschema version.

It is consequently possible to bypass the dictionary altogether in the rendering of a subschema to the DBMS.

- For both data base and nondata base applications, the dictionary does not become involved in the compilation (or precompilation processing) of application programs. For nondata base applications, this means that even though the dictionary may hold "official" FDs, it makes no attempt to ensure that this FD information is in fact that which is used by the application program to reference the described file(s). It is thus possible to create, modify and compile FDs within programs without any use of the dictionary's versions.

For data base applications, data definition is more stringently controlled by the intercession (at compile time or before) of the subschema although, as mentioned above, the definition of the subschema may itself easily bypass dictionary involvement. Thus, although the probability of consistency between program data definition and the DBMS schema version is high (a considerable data base advantage), no additional guarantee of consistency exists with the respective data definition version held within the dictionary.

In other words, even in the data base environment, program and dictionary data definition versions may vary since there is no active dictionary involvement in the relevant compilation events.

The above characteristics of the passive data dictionary role suggest clearly that the burden of control in relevant definitional activities falls heavily to the using organization. If it wishes to ensure that dictionary definitions remain consistent and authentic, then the organization must establish its own practices and procedures for ensuring that dictionary definition versions are never bypassed in any of the system processes mentioned above.

If such bypassing is allowed to occur (a natural though disastrous tendency in deadline-driven system development), then the reliability of the dictionary for where-used reporting and impact assessment is dealt a considerable blow, and the value of the whole dictionary documentaion effort is made suspect.

How can these potential failings be avoided? The necessary practices and procedures inevitably involve rigorous turnover procedures for new and modified systems to operational or production status. The final compiles and linkedits are best performed by the data administration staff itself.

Often a strict coupling, via procedural strategies, is required between source and object libraries and dictionary extraction processing. The content of these libraries must be carefully controlled and coordinated with the dictionary, both directly and indirectly, by the actions of the data administrator (DA).

In short, the passive dictionary places considerable demand for overt and extensive DA involvement in many system support activities in order to ensure the dictionary's viability as an accurate system and data documentation tool.

The Active Dictionary Role

An *active* data dictionary system is one that intercedes in behalf of the using organization to ensure that dictionary content is thoroughly integrated with all relevant system activities. It takes an aggressive role to ensure that it participates in the development of all executable system configurations and that its own content is always current with those definitions used to drive production systems.

Specifically, it ensures that its own "source" data and system definitions are always consistent with those "object" definitions controllng the actions of the DBMS and/or application programs with respect to the data bases or data files of interest.

The active DDS thus remedies the isolation of the passive data dictionary from the compilation processes of the data management complex. By keeping constantly in touch with these processes, it is able to more fully ensure the authenticity of the documentation products it is used to produce, including those for cross-reference and impact assessment needs.

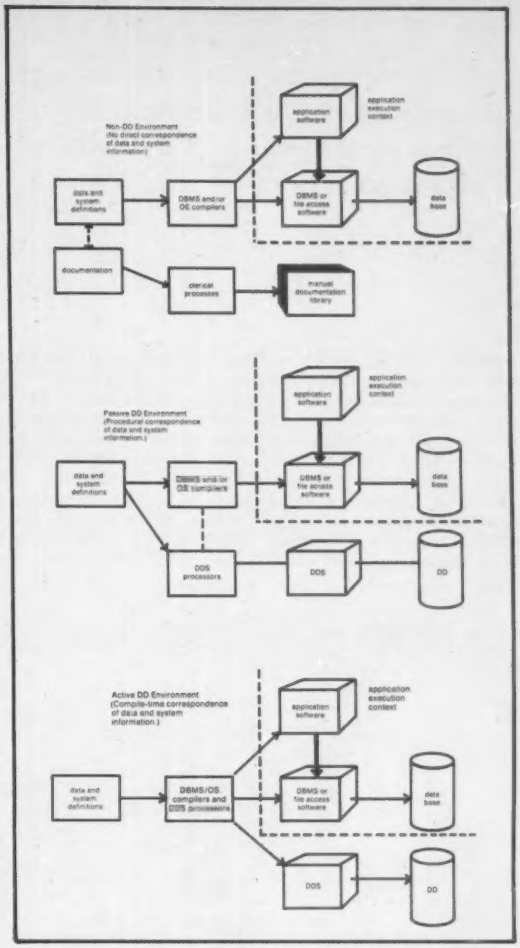


FIGURE 4. DISTINCTIONS AMONG THE DEFINITION PROCESSING SCHEMES OF THREE ALTERNATIVE INFORMATION MANAGEMENT CONFIGURATIONS

Since the active DDS pursues this involvement automatically, it assumes a significant share of the burden of control which, under the passive DDS strategy, falls heavily to the procedural designs the DA must create and enforce.

The active DDS, to fully assume this role, requires partial or complete merging of its own dictionary processing subsystems with processes of other types, in particular the schema and subschema compilers of the DBMS, and the programming language compilers of the host machine's operating environment. Thus, some of the DDS is consolidated with other types of data management processes, though strictly at compilation time rather than during any execution-time activities.

For example, the active dictionary role envisions the involvement of the DDS during the compilation of a user's application program, though not during its actual execution.

The configuration of an active DDS is illustrated in Figure 4, which contrasts the active role with both passive and nondictionary counterparts. Readily apparent in the step-by-step refinement that this figure reveals is the increasing correspondence of object definitions with those documentation products that can be produced by the DDS, as well as steady evolutionary movement of DDS processes toward the execution-time environment of the data management complex.

What other areas of the overall data management complex can be brought into the reach of active dictionary strategies? Although the discussion thus far has centered on data base applications, there is no reason a priori why similar strategies are not possible for nondata base applications. Although subsequent control would be less effective (the absence of the subschema definitional step is crucial), the resulting improvement in dictionary reliability would be substantial.

Other types of system definition processes could also be brought into active dictionary schemes. For example, the generation parameters for the DBMS itself, if contained in the dictionary, could be used on an active basis during DBMS system generation. The same is true for the teleprocessing (TP) monitor generation parameters, as well as other types of definitions for the data communications environment. Broadly defined, such an environment might include both distributed systems and word processing interfaces.

Finally, if DBMS and TP parameters can be contained in the data dictionary on an active basis, there is no reason why the same is not true for the most basic of all definitions in the overall machine complex — those for the operating system itself. Such speculation, of course, takes the active DDS far afield of its original data management role and conceives of its future as increasingly inseparable from the evolutionary development of very basic machine processes.

At this point in time, most commercial DDS are passive rather than active in nature. The strongest potential for the development of an active system, at least theoretically, is where the same vendor supplies all components of the overall data management complex, including the host language compilers, the DBMS (and its compilers) and the DDS. This, of course, normally means the hardware vendor.

A somewhat lesser potential arises where the DBMS and the DDS both come from the same vendor so that rather tight integration of at least DBMS compilation processes is possible. A program preprocessor can to some extent satisfactorily substitute for integration of the host language compiler (unavailable to these vendors), although certain limitations, particularly for nondata base files, are apparent.

The In-Line Dictionary Role

If the active DDS configuration has brought the data dictionary one step closer to the extension-time processes of the data management complex, a number of theorists have wanted to take it even further — to place it directly *in-line* with all execution-time functions. Under such an arrangement, the distinction between "source" and "object" definitions disappears altogether because precisely the *same* definitions used to drive production systems are also used to produce dictionary documentation products.

This, of course, envisions the ultimate guarantee that dictionary documentation is totally authentic. Perhaps more importantly, by completely eliminating the compilation step (a dictionary transaction immediately produces object definitions), the system is made instantly responsive to any and all definitional changes. This would have significant advantages for security and tuning, not to mention the overall administration of the data resources.

There is also speculation that the in-line role would be very useful for the development of distributed processing, where definitional parameters can fluctuate rapidly.

To understand the in-line role, it is necessary to examine the functioning of a DBMS in greater detail. Specifically, the DBMS itself has a dictionary of sorts, more properly called a *directory*, where the object counterparts of source data base definitions are kept. The purpose of this directory, which is managed by one component of the DBMS (specifically, the *DBMS Directory subsystem*), is used exclusively by the DBMS in order to translate and process users' logical requests for data.

Since the definitions it uses for this purpose are derived solely from the data dictionary, where the source is both more comprehensive and more directly usable by the corporate user, the directory may be viewed, in essence, as a shadow image of its "external" dictionary counterpart.

The existence of this "shadow" directory subsystem is illustrated in Figure 5, which is an elaboration of the "active" configuration portrayed in Figure 4. The essential feature of this elaboration is that the output of the DBMS and operating system compilers and DDS processors to the application execution context is shown as being directed to a specific component of the DBMS, the shadow directory subsystem, rather than to the DBMS as a whole.

It is the place of this component that the DDS assumes in an in-line data management configuration. This replacement is illustrated in Figure 6, which completes the steady movement of the DDS toward the execution-time processes of the data management con-

In 1980 over 5000 professionals will attend Q.E.D. courses

Q.E.D. education
The alternative that works . . .

Q.E.D. is the leader in instructor directed education . . . lecture, discussion, workshops, case studies, questions . . . and answers. All the necessary ingredients for effective education. All available in our nationwide EDP education network.

TOTAL PROGRAM
We provide you a total program covering a wide range of disciplines from **systems analysis** and design, **structured methodologies, telecommunications, networking, data base, project management** . . . all developed and taught by senior Q.E.D. instructors of unmatched technical and teaching capability . . . all available at our regional centers or on your premises.

COST EFFECTIVE/DEPENDABLE
To keep the cost down, we pioneered EDPEP, the cooperative, university-based EDP Education Program. EDPEP provides regularly scheduled courses to member companies at under $80 per day. And now our open enrollment policy allows you to register for any course at any location. Our courses are rarely cancelled because they are pre-subscribed by member companies. Over 300 corporations have made Q.E.D.'s EDPEP a major part of their education program.

IN-HOUSE
All our courses are available to you on an in-house basis. If you want a special course or program, we can develop it. Instructor guides, student materials, case studies, workshops, whatever is required. We are ready to assist you.

SEVEN REGIONAL CENTERS
- Babson College, Boston
- Fairleigh Dickinson University, New Jersey
- Drexel University, Philadelphia
- Indiana/Purdue University, Indianapolis
- Southeast Regional Center, Raleigh, N.C.
- The University of Texas at Arlington . . . and
- **Your company's premises**

Take charge of your career. Fill in the coupon and send it to us for details about EDPEP, courses, schedules, books, enrollment, and information about Q.E.D.

Q.E.D. Information Sciences, Inc. • P.O. Box 181 • Q.E.D. Plaza • Wellesley, MA 02181 • (617) 237-5656

For a FREE 100 page catalog with course outlines, schedule & tuition return this coupon

☐ Send Q.E.D. information ☐ Send EDPEP information ☐ *Send book catalog

Name/Title

Company

Address Phone

City State Zip

Mail to:
Q.E.D. Information Sciences, Inc.
P.O. Box 181 – Q.E.D. Plaza
Wellesley, MA 02181
617-237-5656

*See special Q.E.D. book offer in this issue.

"A customer just now committing to DBMS is not, by and large, willing to commit to a DDS simultaneously."

text that were portrayed, in step-wise fashion, by Figure 4.

In this in-line version, the DDS assumes complete responsibility for accepting, validating and translating all data referencing statements passed to the data management software complex by application programs. This, of course, was originally the function performed by the shadow directory subsystem counterpart.

Error in Reasoning

The potentials for enhanced data base support that this substitution offers are unfortunately often misconstrued by observers in the field. What is the error in their reasoning?

It is certainly true that the data dictionary contains more extensive information about data resources than does the typical directory subsystem counterpart. Alias definition, security specifications and edit information are but a few areas in which the information contained in the dictionary is superior to that which is contained in the directory.

By putting this superior definitional complex in-line to the data management system, they reason, it is assumed that substantial capabilities for improved query resolution, tightened security and enhanced data validation procedures could be easily effected.

That much is true. It is, however, entirely possible to achieve the *same* end results through an active dictionary scheme, where these advanced types of refinement in data definition are passed (in object form) from an external dictionary to an "internal" directory. The same capability is thus envisioned with or without an intermediate "compilation" step; that is, with or without an in-line data dictionary.

Thus, many of the advanced capabilities often thought to issue from the in-line configuration arise instead from enhancements to the definitional capacity for data. Whether these enhancements are rendered to the execution-time environment of the data management complex through an in-line DDS or through an appropriately enhanced shadow directory subsystem is a separate and distinct question.

Performance constraints, of course, are much more stringent in the execution environment envisioned by a DBMS than that envisioned by a DDS. It is, therefore, *unlikely* that DDS, as now perceived, will be placed in-line in the near future. Thus, the true benefits of the in-line configuration — total authenticity and instantaneous responsiveness — are unlikely to be realized soon within the data management complex.

For the types of refinement to the definitional capacity for data mentioned above, where realized by enhancement to a directory subsystem, it is possible for the vendor to bypass the "external" dictionary altogether. In other words, the vendor that views the data dictionary in a passive sense (if it occurs to the vendor at all) is likely to move directly toward enhancing the directory subsystem without worrying about how these elaborated data definitions will find their way, in a consistent and timely fashion, into a data dictionary.

In so doing, these vendors are not only guilty of repeating past mistakes (the above scenario, with slight changes in wording, could be applied to the very origin of DDS in the early 1970s), but of forfeiting the very considerable advantages of the active DDS configuration as well.

In fairness to vendors, however, it must be noted that since the customer base for DBMS lags by several years in its awareness of DDS, the commercial pressures toward keeping the DDS "separate," even if misdirected, are nonetheless relatively strong. A customer just now committing to DBMS is not, by and large, one which is also willing to commit to a DDS simultaneously. As awareness of data resource management concepts grows in the DP community, however — and there is evidence of substantial progress in this direction — this traditional difficulty may become less and less a factor.

Upon arrival at the in-line dictionary configuration, the distinction between

A World of Value

Join those around the world who have invested with success in Systems Software from Computer Associates

Of course, CA products quickly pay for themselves. However, it's the ongoing benefits they provide that make them so successful and appreciated by our 8,000+ customers world-wide.

Our policy of continuing enhancements to these products means you'll never be tied to last year's model. You're guaranteed a "State-of-the-Art" system for years to come with today's investment in Computer Associates Software.

Our high-spirited professionalism, and pride in serving you, makes us unique and pleasurable to do business with. Take this opportunity to contact us and see what we mean. You'll be on your way to a wise and successful investment.

CA-JASPER+
A comprehensive job accounting system for efficient performance evaluation, cost analysis and billing.
CA-EARL
A fast, flexible system for information retrieval that reduces programmer workload.
CA-VM/370
• CA-SORT/CMS • CA-IDOS/VS
• CA-VM/SAM • CA-SYMBUG
CA-SOLVE
An interactive software system that revolutionizes COBOL program testing and debugging.
CA-SORT
A unique sorting system that runs faster, uses less resources, gives you greater throughput...
1979 datapro HONOR ROLL
CA-DRIVER
A multi-faceted Job Execution Management System.
CA-DYNAM/D
A system for optimum disk space utilization with catalog support.
CA-DYNAM/T
A tape library control system that assures optimum security, greater throughput and better use of system resources.
CA-DYNAM/FI
A file independence facility which allows the user to utilize all files freely without the need to modify source programs.

Member Software Industry Association

HEADQUARTERS
Computer Associates Building 125 Jericho Turnpike Jericho, NY 11753
Offices in the United States, Europe, Australia, Canada and Japan. Distributed in 32 countries world-wide.

MAIL THE COUPON OR CALL TOLL-FREE 800-645-3003 NY State Call: (516) 333-6700 CANADA Call: (416) 842-2110

COMPUTER ASSOCIATES

125 Jericho Turnpike, Jericho, NY 11753
YES! Send FREE information as I've indicated:

Name _____
Title _____
Company _____
Address _____
City _____ State _____
Zip Code _____ Phone _____
CPU _____ OP/SYS _____
CW 9/17/80

← CHECK OFF FOR INFORMATION REQUESTS.

CA-SORT	CA-DYNAM/D	CA-DYNAM/T	CA-DYNAM/FI	CA-JASPER+	CA-EARL	CA-SOLVE	CA-DRIVER	CA-VM/370	
■	■	■	■	■	■	■	■	*	Increase Programmer Productivity
		■		■	■	■	■	*	Improve Human Resources
■	■		■				■	*	Save Disk Space
■		■	■					*	Save Tape Drives
■	■	■	■	■				*	Improve Elapsed Time
	■	■		■	■		■	*	Increase Management Control
	■	■	■	■	■		■	*	Improve Security
■	■	■	■	■	■	■	■	*	Improve Operations Throughput
■	■	■		■		■	■	*	Save CPU Time
	■	■		■	■	■		*	Save Supplies

* SEND FOR DETAILS ON MULTIPLE PRODUCT BENEFITS.

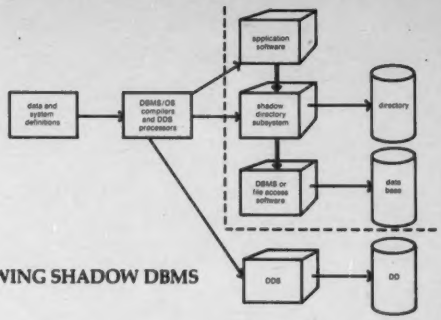


FIGURE 5. ACTIVE DDS SHOWING SHADOW DBMS DIRECTORY SUBSYSTEM

DDS and DBMS, relatively clear-cut until this point, now becomes increasingly indistinct. In fact, it may become more correct to speak of a single unitary software construct — a DDS/DBMS — rather than two distinguishable and separately operable components.

The tables have now been completely turned. With the in-line configuration, the dictionary is no longer a passive spectator of the data management game, but indeed one of its most central players. Responsibility for data base definition and control — the quarterbacking in this game — lies with the DDS, a flashy rookie, rather than the DBMS, the end of a long-standing trend. The DDS' position is unexpectedly, and perhaps ironically, ascendent.

PART III
Feature Development in DDS: A Review

The purpose of the following section is to suggest some of the areas into which DDS may be evolving, based on trends visible in the industry.

On the technological horizon are a number of innovative DDS feature developments that envision new types of processing and interface subsystems. Some of these developments include:

• *Program library management.* By using dictionary models of jobstreams for program compilation, link-editing and execution, along with version and security information contained for programs in the data dictionary, the DDS has sufficient capabilities to combine forces with a program source and object library management system to tightly control the modification of programs, as well as the configuring of executable systems. The exchange is reciprocal — extensive information about source and object programs can be gathered for the dictionary by the DDS' involvement in these areas.

In effect, this very significant administrative capability represents an important enhancement in the DDS' ability to participate "actively" in various system processes.

• *Test data base generation.* Using schema and/or file definitions in the dictionary, a DDS utility can be instructed to transfer data from existing data bases and/or files into new ones, specifically and exclusively for testing application programs. This assistance helps eliminate the chance of contaminating "production" data bases.

• *Data flow and structure diagrams.* By interfacing dictionary information to mechanical plotters, data and program flow diagrams as well as schematics of data model complex occurrences can be produced.

• *Data entry processors.* Since the dictionary includes information about the files that have been established to

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1343 DALLAS, TEXAS

POSTAGE WILL BE PAID BY ADDRESSEE

UCC
UNIVERSITY COMPUTING COMPANY

Software Division
Marketing Dept.
P.O. Box 47911
Dallas, TX 75247

COMPUTERWORLD

Fill out and mail this self-addressed postcard.
Please send me more information on:

- ☐ UCC ONE (Tape Management)
- ☐ UCC TWO (DUO, DOS under OS)
- ☐ UCC THREE/ADAM (Disk Management)
- ☐ UCC SIX (PDS Space Management)
- ☐ UCC SEVEN (Automated Production Control)
- ☐ UCC TEN (IMS Data Dictionary/Manager)
- ☐ UCC ELEVEN (Automated Recovery and Tracking)
- ☐ UCC RELIABILITY PLUS (Hardware Vendor Accountability)
- ☐ UCC GENERAL ACCOUNTING SOFTWARE
- ☐ UCC BANKING SOFTWARE
- ☐ Other Information _____

Name/Title _____

Company _____

Address _____

City/State/Zip _____

Telephone _____

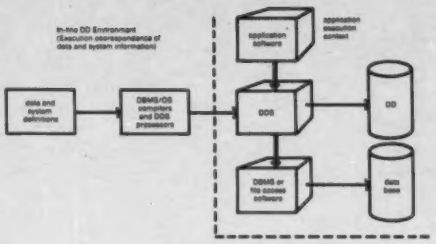


FIGURE 6. THE DEFINITION PROCESSING SCHEME OF THE IN-LINE INFORMATION MANAGEMENT CONFIGURATION

receive the given data, a utility subsystem can be built to process and transfer the input data. By informing the dictionary of edit rules for the data, the data entry processor can additionally support validation functions for the input.

• *Word processing interfaces.* Using dictionary information for defining and supporting manual forms (or their equivalent) and a corporate glossary, the DDS can be coupled with word processing systems to perform various types of prompting, editing, storage and forwarding.

Additional coupling with electronic mail systems and reprographic devices puts the DDS squarely at the center of the development of the so-called "office of the future."

Some other relevant observations include the following:

• Experience with DDS and its use for the control and enforcement of standards and conventions shows conclusively that the sooner the involvement of the dictionary with new systems, the more effective the eventual support will be. This strongly suggests that regression of DDS involvement to the system design phase has substantial potential for improving administrative control.

• Changing economics within the DP industry have begun to make people more expensive than machines. Since the system design area remains one of the most people-intensive parts of the DP enterprise — and since there now exists very little systemized support for this area — any automated tool with potential for boosting productivity in this area is being eyed carefully.

• New extensibility features in DDS have improved the flexibility of DDS by an order of magnitude of more. This feature is optimal for areas where execution performance is not a crucial factor. The system design phase is just such an area.

The clear implication of these observations is that movement to exploit the DDS for the preimplementation phases of the system life cycle is inevitable.

In spite of these rather clearly delineated trends, the DDS' encroachment into the system design area is likely to be gradual — just as for any other major innovation that occurs in data processing.

Even so, the eventual outcome of this movement may radically alter the way in which applications are now conceived and implemented.

Why is this so? Once a DDS is invested with information about the *design* of systems, the next obvious step becomes assistance for the *implementation* of those designs — in other words, DDS assistance for automatic system installation.

A development of this nature would represent the most significant change in the method of implementing computerized systems on a mass, commercial scale since the widespread acceptance of Cobol two decades ago. □

Is a wild IMS loose in your shop? Get *all* of it under control with UCC-10.

There is still only one dictionary package that controls the entire IMS environment — the UCC-10 Data Dictionary/Manager.

UCC-10 controls the online blocks — MFS and data communications — as well as the databases and applications. It's the only dictionary package that does this. Which means that — in an IMS recovery situation — you can be back online in a few minutes, instead of a few hours.

UCC-10 also allows you to build the IMS database the way it will work best for you. This cuts program design and development time significantly, not to mention eventual running time.

If you've got an IMS control problem — solve the whole problem, not just half of it. Call us about UCC-10 today at 1-800-527-5012 (in Texas, 214-353-7312).

And, while you're at it ask us about:

A Tape Management System that protects data from loss or destruction (UCC-1).

A DOS Under OS System that lets you execute DOS programs without conversion (UCC-2).

A Disk Management System that can save the cost of new disk drives (UCC-3).

A PDS Space Management System that eliminates PDS compression (UCC-6).

A Production Control System that makes scheduling systems obsolete (UCC-7).

An Automated Re-run and Tracking System that solves your re-run problems (UCC-11).

A Hardware Evaluation System that gives you the facts on reliability (UCC Reliability Plus).

General Accounting software packages.

Application software for the Banking and Thrift industries.

UCC

SOFTWARE AND COMPUTING SERVICES

UNIVERSITY COMPUTING COMPANY • DALLAS • LONDON • TORONTO • ZURICH

UCC is a subsidiary of Wyly Corporation (NYSE)
For information and an Annual Report, write 1000 UCC Tower, Exchange Park, Dallas, Texas 75235

By Thomas F. Meurer

SOLVING THE MYSTERY OF DATA BASE DESIGN

Data base design, from the beginning, has been a mystery to nearly everyone. Regardless of the data base management system (DBMS) being used, there are few strict rules in logical and physical design.

When a data-based system goes into production, there is usually little documentation of how individual data items are related to one another and why specific physical design choices have been made. This causes great difficulty when the data administration staff grows, involving more people in the tuning and maintenance effort, and when the data base content grows and the data administration staff is required to integrate a second application with the first.

Data base designs cannot just be models of the data relationships in the company. They must be tuned to the particular business functions being executed by the processes in the programs that reference the data base.

This article will present a structured data base analysis and design methodology that is the result of several years' experience in helping users develop effective data bases. This methodology establishes a baseline data design that is independent of hardware and software and that forces documentation of logical data relationships and the reasons for particular physical design choices.

Structured Methods

In the last few years, structured analysis, design and implementation techniques have gained wide acceptance. These techniques go hand-in-hand with the structured data base analysis and design methodology being described in this paper.

Figure 1 outlines the structured project life cycle. In the structured project analysis phase, business analysts study and model the business information flow using a top-down approach and pictures rather than text to describe that flow.

The analysis phase produces a structured specification consisting of data flow diagrams that identify processes, data stores and data flows among them. Each process is completely defined with structured business logic. This logic can take the form of decision tables, decision trees, psuedocode or tight English.

In order to produce these diagrams and definitions of the business, codification of business logic that usually occurred in the programming phase in classical development techniques must be done in this analysis phase. A data dictionary is required to support structured analysis because of the volume of data that must be recorded and cross-referenced.

Structured design follows analysis and determines the hierarchy of systems, subsystems, programs and modules to support the structured analysis specifications. Processes defined in the structured analysis are assigned to modules in the design phase.

Structured implementation follows design and includes top-down programming and testing techniques that physically implement the structured design specifications.

The combinaton of structured application development techniques and the use of data base technology gives a project life cycle profile as shown in Figure 2. Studies show that the project life cycle time is shorter using structured analysis and data base techniques. Fewer lines of code are written and major flaws are less likely to be found in the analysis or design at implementation time.

But these are small factors compared with the results of studies that show that systems implemented with structured techniques and data base technology require approximately one-third the maintenance time, effort and cost of systems that do not use these techniques. In most installations today, more that 50% of the DP dollar is spent on software (as opposed to hardware) and more than 50% of the software portion is spent on maintenance of existing systems. There is, therefore, tremendous benefit to many companies in adopting both of these methodologies.

These methodologies treat data as a corporate asset. They identify and define data elements in the business, identify dependencies and relationships among those data elements and identify how the company uses each data element.

Gane and Sarson have stated that data flows are data structures in motion and that data bases are data structures at rest. Data bases must be designed to support the frequency and turnaround time of all data flows in the business within the hardware/-software/operating system constraints.

Enterprise Surveys

A project is concerned with only a portion of the business in the enterprise. At the outset, a broader view of the enterprise is required from two perspectives:

- Identification of major processes throughout the business and the identification of the major inputs and outputs of each process.
- Identification of the major information centers in the business and their relationship to one another.

These enterprise views are required for many reasons. Automatic or manual systems already exist in every part of the business. Any new system (automated or manual) replaces a portion of that current information flow network and, therefore, must interface properly with each current system that feeds information to the new system or accesses information from it. We must know those interfaces prior to structured analysis.

Likewise, any new files or data bases developed by data administration in the support of new project development either replace or duplicate data structures at rest that will be used to support the business information network. Many relationships are inherent in the files and data bases based on the way the company does business. These relationships must also be identified at a level in the enterprise prior to any logical data base design effort.

Figure 3 depicts both of these enterprise survey efforts. These efforts can be done in a week or two of effort to the degree of detail necessary to support the follow-on steps on both the systems development side and data administration side. The global data flow diagram and data structure diagram developed by the surveys are such that they:

- Are used extensively in communications at all levels in the business.
- Do not change unless the company makes a major change in the way it does business.
- Provide focus on areas of development and maintenance.
- Can be color-coded or highlighted to identify past, present and future efforts.
- Act as a reminder of major processes that are candidates for sharing data.

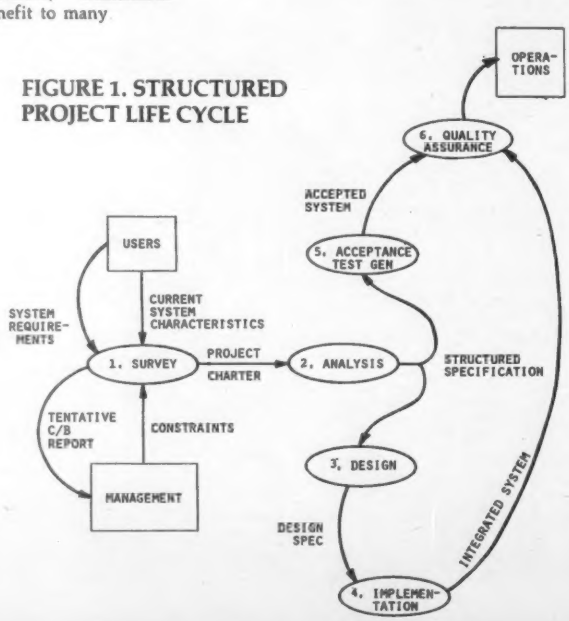
Figures 4 and 5 are examples of a global data structure diagram and a global data flow diagram, respectively. On the global data structure diagram, the boxes represent information centers and the arrows represent lines of relationship. The arrows carry a notation showing whether the relationship is a one to one (1-1) relationship, one to many (1-N) relationship or many to many (M-N) relationship. We use the same standards for numbering information centers in the global data flow diagram that are used in numbering processes in data flow diagrams.

Both of these diagrams must be contained on one sheet of paper. There is no requirement about how large the sheet of paper may be or how small the printing on them can be. Typically, these diagrams take the form of wall charts.

Once their accuracy has been validated by walkthroughs with the users, they should be sent to the printer and high-quality, poster-type charts produced with color coding to highlight sections or interfaces of major interest. These charts are used frequently in conversation, planning and education.

The data base designer uses the

FIGURE 1. STRUCTURED PROJECT LIFE CYCLE



Thomas F. Meurer is president of Eastern Technical Associates, Inc., a consulting firm in Newton, Mass.

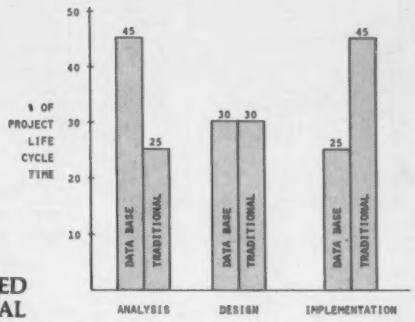


FIGURE 2. STRUCTURED DATA BASE VS. TRADITIONAL

global data structure diagram (DSD) along with the project data flow diagrams and data dictionary reports in a project entity-relationship (E/R) modeling step to produce a logical data structure diagram. This step with its inputs and outputs is shown in Figure 6.

stored by a data flow is used by at least one other data flow.

- Verifying that each data element needed to support a data flow has been stored by some other data flow.

Do not worry about differences in the information centers identified in the global DSD and the data stores identified in the project data flow diagram or between the global DSD and records identified in this logical data base design step. These differences will be corrected in the next step.

A logical data structure diagram is shown in Figure 8. The blocks indicate logical records which are (adjusted) data stores from the data flow diagrams.

All access keys to a logical record are listed; these are known from the detail of how a process in a data flow diagram must access a data store. All record relationships are shown; these relationships represent a further investigation of the data relationships shown in the global data structure diagram for the collection of data elements that the project addresses. And, finally, relationships are again labeled as 1-1, 1-N, M-N.

In the past, many data base designers have gone directly from a logical data structure diagram similar to Figure 8 to a physical design process. However, several problems may remain from a data administration point of view in the logical DSD. Specifically, it might contain:

- Repeating groups.
- Duplicate data items.
- Nonunique record occurrences.
- Levels of data element dependencies. In addition, this logical DSD is not necessarily tuned to the requirements of the processes defined in the data flow diagrams. In short, the record contents may not match the way the company actually deals with the data and the record formats themselves can be difficult for end users as well as programmers to deal with.

Therefore, this logical DSD is a temporary DSD for use as an interim working document by data administration only. It is input to the following step, called normalization.

FIGURE 3.

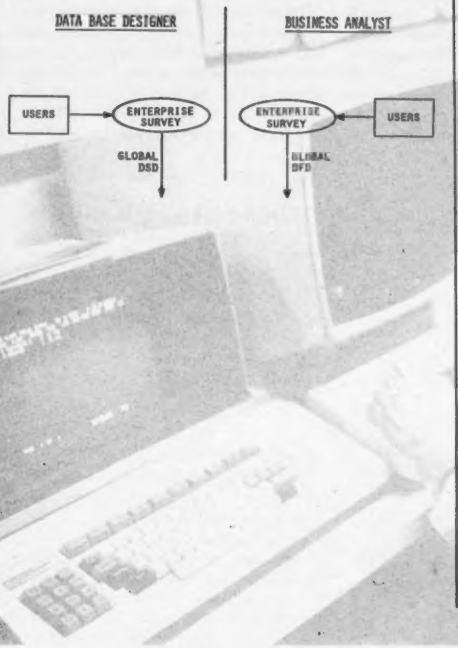
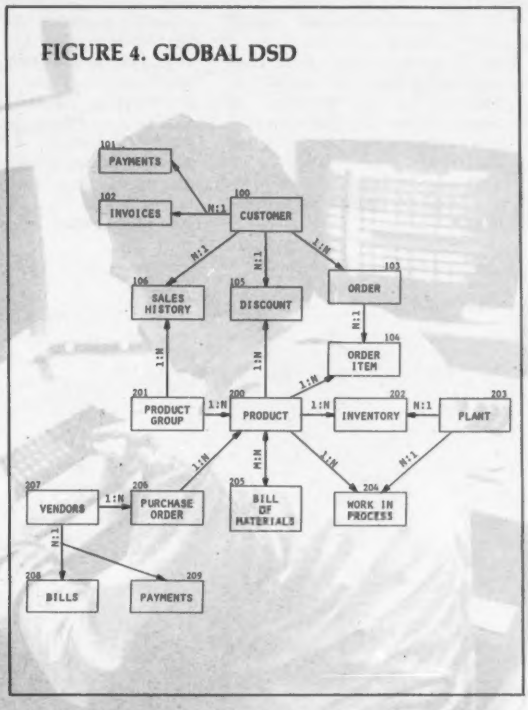


FIGURE 4. GLOBAL DSD



The Normalization Step

Normalization is simply a study of the data that an organization uses and relationships and dependencies among that data. The output of normalization is a set of data table definitions which are provided to the project structured design process and are used by the data base designer in the definition of physical data structures and the construction of subset views. Figure 9 shows the normalization process.

Normalization began through the efforts of E.F. Codd of IBM toward a precise mathematical treatment of data definition and processing. Codd has done much valuable work in this area, but his work is described in his papers in such a way that few people can understand it.

Chris Date, also of IBM, has restated much of Codd's work in a language that we can understand, and he has also done additional research work of his own in this area. The normalization process can be characterized by the following:

- It is derived from a strict set of rules - there is no room for interpretation .
- It dictates no particular physical arrangement of data.
- It identifies duplication of data elements and uses that duplication to identify relationships among normalized records.
- It identifies all dependencies among data elements.
- There are several levels of normalization and the normalization process is 100% reversible through those levels.
- Normalized records are 100% independent of the process and data flow requirements identified in data flow diagrams.
- It forces unbundling of concepts such as the attributes of record relationships in most DBMS (e.g., membership and order options.)

The Rulebook

Levels of normalization are defined according to the following rules: A record is in first normal form if — and only if — it contains no repeating groups. If records in the logical data structure diagram contain repeating groups, those groups must be removed from that record and stored as another record carrying the minimum number of additional data elements besides the repeating groups that are necessary to form a key that uniquely identifies each group occurrence.

A record is in second normal form if it is in first normal form and every nonkey element is fully dependent on the primary key. If an element is not a key element in some other record, determine the element or elements on which it is dependent and add the element(s) to that record. Form a new record type if necessary.

A record is in third normal form if it is in second normal form and every nonkey element is fully dependent on the entire primary key and no nonkey

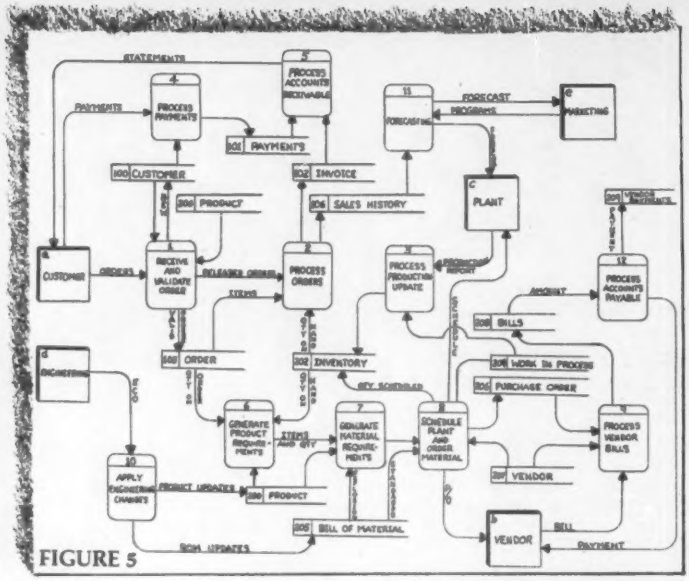


FIGURE 5

element is dependent on any other nonkey element. If there are such multiple dependencies among the elements in a record, remove those nonkey elements and, following the rules in the second normalized form paragraph above, relocate them or form additional records if necessary.

Figure 10 depicts the universe of record definitions (relationships) and their nesting effect.

Fourth, fifth and sixth normal forms have been defined, but they are of lesser importance to data administration from a data base design point of view than the first three.

The normalization process consists of using the above rules to decompose the logical data structure diagram into its normalized form (Figure 11). Lines of association are drawn from the primary key of a record to each record in which a data element which participates in that key is duplicated. Each of these lines of relationship is a one to many relationship.

A portion of the global data structure diagram represents the parent diagram according to structured leveling techniques to this normalized data structure diagram. At this point, then, adjustments may need to be made to the global data structure diagram to properly represent that parentage.

Likewise, the records in the normalized data structure diagram are numbered according to the structured leveling numbering scheme to show many normalized records may be derived from a single parent. Note from Figure 9 that some normalized records may have been defined already by data administration from prior efforts. This collection of normalized records is used as a starting point.

For each new effort, some data elements in the project data flows will be in the existing normalized records and should be properly located in those records. Some of the data elements in the current project data flow diagrams will be added to existing normalized records, and some will form new normalized records.

At the end of the normalization process for each project, the complete set of normalized records should be cataloged in the data dictionary and reports distributed to all interested data processing and user personnel.

The benefits of normalization are:

- It provides the simplest possible representation of data.
- The representation is the simplest to understand and deal with.
- All file access methods and DBMS can represent normalized records.
- The physical implementation can be changed from one set of file structures to another without impacting the normalized record definition.
- They are the foundation for relational data base systems, which are definitely coming.
- They allow the use of relational operators to define manipulation of the data stores in the structured design phase.
- They represent the perfect place to start the physical data base design process. The designer is aware of all compromises and adjustments made to the normalized records in the physical implementation.

Figure 12 shows the universe of data manipulation language statements in the three major data base models in existence today. Relational data base systems have fewer DML statements than hierarchical systems, which in turn have fewer DML statements than network systems.

Since data processing in any organization might use one or more of these data base architectures, the normalized records and the smallest set of relational operators on those records form the only common baseline definition for the company's data structures and data flows.

It is possible for the structured designer to use a set of relational operators to specifically state how the normalized records are accessed by the data flows. These specifications become part of the structured design specification and can be stored in the data dictionary, where they are available to other application programmers.

Using the normalized data structure diagram, the structured designer adjusts the data flow diagrams as necessary to deal with the normalized records rather than the original data stores. The designer can then design systems and programs that deal with the normalized records. As a result, the design is independent of the physical structures chosen by data administration.

In the physical design process, there is still freedom to choose whatever storage structures are appropriate including sequential, indexed sequential, Vsam, one or more DBMS and a centralized or distributed environment.

The physical data base design is concerned with the definition of the physical storage structures to allow the normalized data structures to support the frequency and turnaround time of all data flows within the constraints of hardware, software, operating system and processing cost. The data dictionary must contain estimates for the frequency and turnaround time of each data flow for this project.

If other normalization processes and physical data base designs have been done by the organization before the current project, then the data flow requirements of the current project must be merged with those existing data flow requirements that caused the particular physical design to be chosen. The physical design is then reconstructed.

Figure 13 depicts the input to the physical design process. The particular physical design options are dependent upon the file structures and DBMS that is installed at the user site and the design and data management tools that can be used with it. From the normalized data structure diagram and the service-level requirements defined for the data flows, the designer has all the information that is needed to design physical structures to support those data flows.

The design steps include, but are not limited to, the following:

- Determining record formats and mappings to support the data groupings.
- Determining physical data element

FIGURE 6

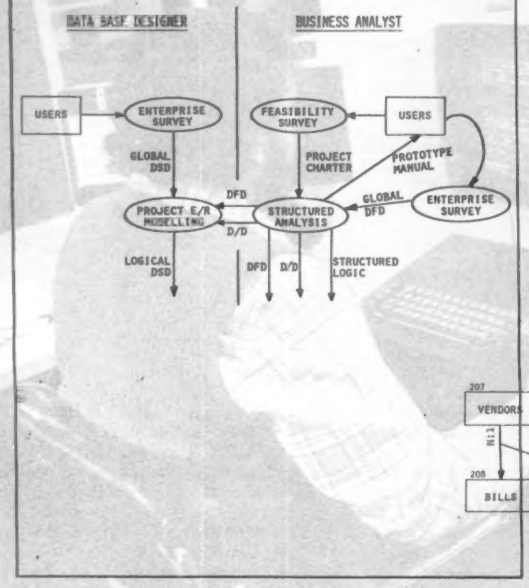
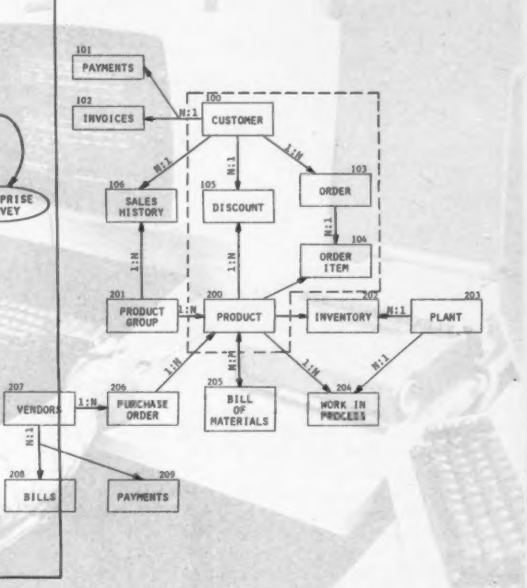


FIGURE 7. ORDER ENTRY PROJECT



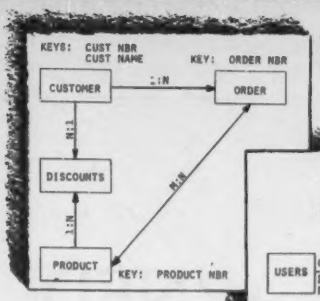


FIGURE 8. THE LOGICAL DATA STRUCTURE DIAGRAM (RESULT OF E/R MODELLING)

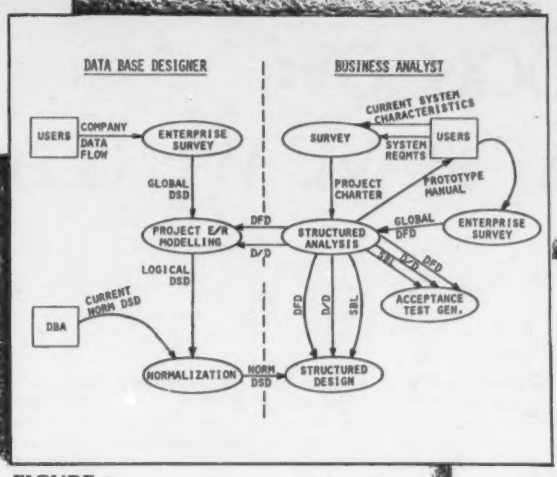


FIGURE 9

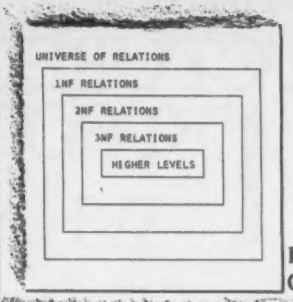


FIGURE 10. UNIVERSE OF RELATIONS

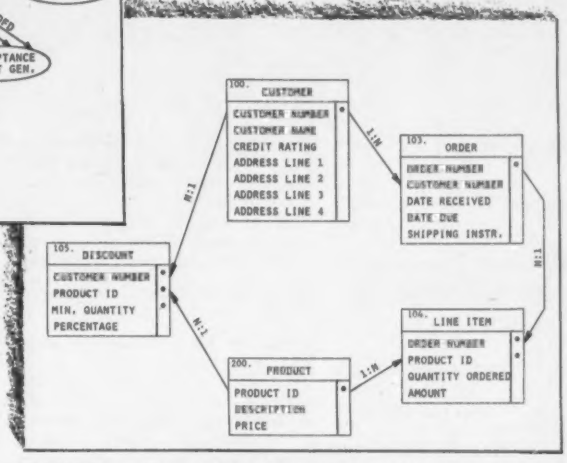


FIGURE 11. THE NORMALIZED DATA STRUCTURE

names and attributes.

- Selecting primary key access mechanisms and physical placement mode.
- Selecting alternate (secondary) key access mechanisms.
- Selecting mechanisms for record clustering.
- Determining requirements for distributed processing/distributed data bases.
- Introducing planned redundancies.
- Planning the use of the data management facilities that accompany the DBMS including report writer facilities, query and update languages and data entry and validation facilities.

All deviations from the normalized data structure diagram that are made to specifically meet the service levels defined in the data flow diagrams can be easily recorded. Therefore, the data base design function has all of the information it needs to maintain the data base structures as the service levels change, as new services are added and as additional applications are implemented which share data elements included in the physical design.

Constructing Subset Views

The purpose of subset views is to make the user or programmer as independent as possible of the physical characteristics of the data base design. The more the physical data base design deviates from the normalized data structure diagram, the less likely it is that the subset view is independent from that physical design.

One test of a DBMS' data independence is the ability it has to mask physical attributes about the data structures from the programmers and end users. No DBMS does this completely, but each offers some useful features in this area.

Figure 14 now represents the complete structured data base design process with the information flow shown between the steps in the project life cycle and data base design. The input to the construction of subset views is the physical data base design, the normalized data structure diagram and the structured design specifications which define the date element needs of each program.

Subset views are then constructed as allowed by the DBMS to provide access to just those data elements needed and in such a way that the physical placement of those data elements is not (necessarily) known to the requester.

The output of this process of constructing subset views is printed or written documentation of all information needed by the programmer to write, debug and maintain the program. This again varies with the particular DBMS.

The methods outlined in this paper have been developed for the following reasons:

- To produce a businesswide view showing data at rest in major information centers (global DSD) and in motion supporting major processes (global DFD).
- To produce a baseline normalized data design for each project that is independent of company processes and physical design considerations.
- To construct a physical design that is as close as possible to the normalized design with adjustments to support the data flow requirements and to fit within data processing limitations.
- To construct subset views that mask as much of the physical design from programmers and end users as possible.
- To force the production of documentation at each step so it is clear what the data relationships are and why design choices were made. This documentation is crucial to the follow-on quality control and maintenance efforts.□

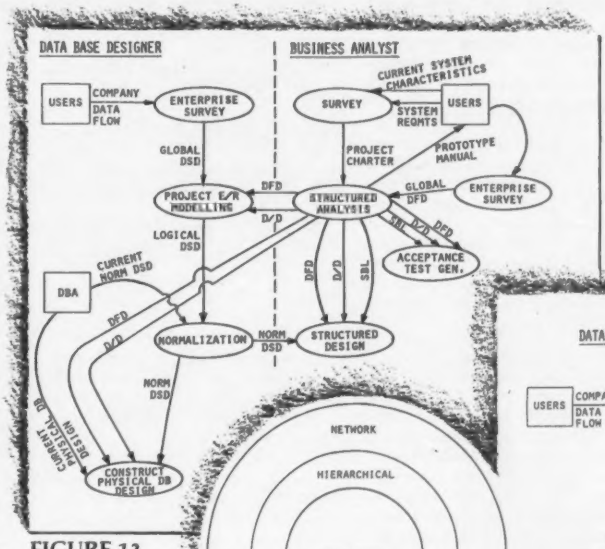


FIGURE 13

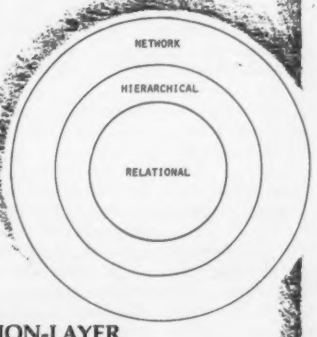


FIGURE 12. ONION-LAYER CHARACTERISTICS OF THE THREE DATA MODELS

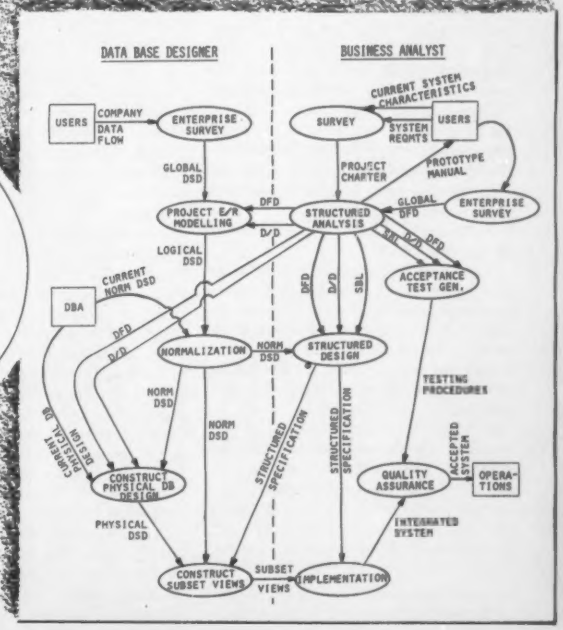


FIGURE 14

By Chris Gane and Trish Sarson

STRUCTURED METHODOLOGY: WHAT HAVE WE LEARNED?

We really get the users enthusiastic about a project with data flow diagrams.
— Oil company system development manager

Using hierarchical structure charts, we made changes five to 10 times faster than with traditional flow charts.
— Revenue accounting project leader

Programmers can be productive on a project sooner because the structured logic package is more complete than the traditional spec.
— Aerospace system designer

We found fewer problems during testing because the structured design document is itself testable.
— Programmer on purchasing project (IMS)

Over the last three years, a complete set of structured techniques — structured analysis, structured design, structured coding, top-down development and structured walkthroughs — has come together into a coherent structured methodology. Firms that have used the methodology on projects ranging in size from three man-months to (in one case) several million dollars are repeatedly reporting results like those described in the actual quotes above.

We asked Carl Garrison, manager of information systems for The Superior Oil Co. in Houston, what his experience with the structured methodology had been to date. "We've used the complete set of techniques over the past two years" Garrison told us. "Some projects we've gone all the way through with the structured techniques; our largest project has sections that have used structured analysis and structured design and some sections that are now in programming."

We asked Carl to tell us more about this large project.

"It's a petroleum production revenue and accounting system, with a development budget of several million dollars. It will track the production of oil and gas for Superior, from the time it comes out of the ground to the time we pay royalties on the revenue," he explained.

"I think the major impact of the structured techniques has been on the communication with the users and on the user involvement we've gotten. Every one of our users was very impressed with the data flow diagrams we showed them. To our surprise, some of them were even prepared to read the hierarchical structure design charts and tell us where we'd missed a business function!

"We've also had very good results with the changeability of our designs and the changeability of the resulting code."

Chris Gane and Trish Sarson are the principals at Improved System Technologies, Inc., a consulting firm in New York.

Structured Analysis

What are the techniques to which Carl referred?

The key to successful structured analysis, it now appears, is the building of a graphical, logical (in the sense of "not-physical") model of the required system. Such a logical model, plus a statement of systems objectives and system constraints, constitutes an adequate requirements statement, which has two virtues.

First, it expresses what the system is required to do, while not making any commitment as to how the system should physically be implemented. Thus a systems analyst can use the logical model to express system requirements without having any detailed current knowledge of DP techniques and present the model to the systems designer — leaving the designer in no doubt as to what the system should do, but giving him the greatest possible freedom to devise a physical design which does it most cost-effectively.

Second, the logical model turns out to be a very good way of showing nontechnical users what the nature of the system is going to be and how the various parts will fit together.

Logical Model

What does such a logical model look like? Let us consider one of the simplest of all businesses, Joe's Auto Parts (slogan: "If you want it, we'll get it"). Joe holds no inventory, but takes orders for spares from his customers, promising future delivery. When he has enough orders for spares from a given car firm to earn a bulk discount, he sends the firm a bulk order and, on receiving the bulk order, gives each customer what they want, charging them the full retail price.

Figure 1 shows a very general picture of the data flow in Joe's business. Customers send in orders, and the single process block processes them, using a store of data (in some form) about parts details — price, delivery and so on — and a store of data about customers — whether they are in good credit standing, and so on.

This very crude data flow diagram (DFD) uses just four symbols, as shown in Figure 2:

- An external entity symbol (square): a source and/or destination of data outside the system.
- A data flow symbol (arrow): a pathway along which data moves into, around and out of the system.
- A process symbol (rounded rectangle): some function which transforms data.
- A data store symbol (open-ended rectangle): a place in the system where data is stored in some way.

It turns out that these four simple symbols are the minimum required to model a commercial system and that these same symbols can be used to model at any level of detail. For instance, Figure 1 is clearly rather trivial — let us expand "Process Orders" to see more detail.

Figure 3 shows orders coming from customers into a more detailed process "Verify order is valid." This process needs data about dataparts (to check that the part exists and is available) and data about customers (to check credit).

Out of the process come valid orders — one important simplification is that in high-level DFDs, we omit error paths and exception handling. The valid orders are stored in a data store of "Pending Orders" which may be a spike on the office wall or may be an IMS data base (we do not care at this stage). We just note that there is a logical requirement to store valid orders until they can be batched to make a bulk order.

The lower stream of data in Figure 3 shows the data describing the bulk shipment from the suppliers being verified (note there are no error paths shown), assigned to individual orders and delivered to customers.

Clearly, there must be some financial flows. We must bill our customers and pay our suppliers, as show in Figure 4.

Figure 4 — a high-level logical data flow for transaction processing at Joe's Auto Parts — has several interesting features:

- It is logical, so it makes no distinction between manual processes, computer processes, tape files, disk files or any such technicality.
- Because it is logical and uses a very simple symbology, it is readily understood by non technical users, such as Joe. User managers commonly say, when such diagrams are explained to them, "That's the first thing you computer people have ever shown me that I have understood!" Or, "Now I know what that system does!"
- Though abstract, it is precise enough to be criticized. "I see what happens to payments from my customers, but what happens if they don't pay?" Joe might ask.

We would have to admit that we are missing functions to analyze accounts receivable for delinquency and send out dunning letters to customers. Of course, we are glad to find out that we are missing a desired function before we start to design and program. Although much more complicated for a real system, this is the type of data flow diagram with which Carl Garrison has had such good results.

Greater Detail

If still more detail of a process is required, we can "explode" the process box into a more specific lower-level DFD. Figure 5 is an example of such an "explosion" of the "Assemble Requisition to Suppliers" process from Figure 4.

If more detail of each data flow is required, we need to be able to express the logical nature of, say, "Orders." Since a data structure like this is hierarchical, we may give a series of more and more detailed answers to the question "What do you mean by orders?" First we may name the chief components:

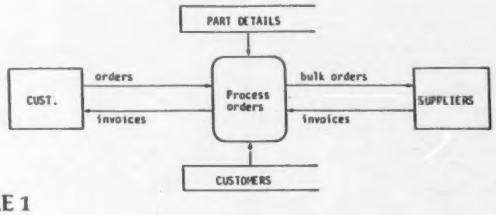


FIGURE 1

```
ORDER
    ORDER-IDENTIFICATION
    CUSTOMER-DETAILS
    PART-DETAILS (iterated)
```

If that is not detailed enough, we can state the subcomponents of each component, as shown in Figure 6.

We express the data structures and data elements with honest, meaningful names chosen by the analyst. When we get down to the level of a data element (a piece of data that cannot be meaningfully subdivided), we must specify its logical (not-physical) nature. Thus we might express the North American telephone area codes as in Figure 7.

If all data structures are composed ultimately of data elements and if we can (a) define each data element, (b) state the way in which they are combined into data structures and (c) state which data structures move along the various data flows and which are found at rest in the data stores of our DFD (Figure 8 diagrams these relationships), then we have provided all the data objects required for a logical data dictionary.

The logical data dictionary is the place where all detailed definitions of data objects are stored.

Defining Data Stores

Now we have to consider the definition of data stores. For every data store we must define its contents (in terms of the data structures defined in the data dictionary). In addition, for some data stores, we will have to show the immediate accesses that have to be made to it.

The definition of data store contents is in principle very simple: What comes out must go in — no more, no less.

So we look at the data flows coming into the data store, compare them with the data flows coming out, make sure that nothing is missing and that nothing is redundant, then derive the data store contents accordingly.

Suppose we had detailed data flows for Joe's Auto Parts and the complete flows in and out of "Pending Orders" were as shown in Figure 9. Assume that the detailed contents of each flow have already been defined in the data dictionary. We have some nonimmediate accesses to data (batched orders, orders for a given part) and some potentially immediate accesses which may physically be on-line inquiries: "What orders are there for Supplier X?" "What is the status of Customer Y's order?" and so on.

What must the contents of the "Pending Orders" data store be? Since it is fed by a stream of orders and streams of orders come out, it surely must contain records of the data structure "Orders," defined as in our previous list.

Is that enough? Inspection shows us that it would be. Of course with a more complex data store, with say 10 inflows and 20 outflows, the derivation of contents is far from being so easy, and the detailed logical components of each flow must be compared. It is valuable to reduce the contents of each data store to several tables, or relations, in third normal form, thus removing redundant data and getting the structures as simple as possible:

```
ORDER
   ORDER-IDENTIFICATION
      CUSTOMER-ORDER-NUM
      CUSTOMER-ORDER-DATE
   CUSTOMER-DETAILS
      CUSTOMER-NAME
      PERSON-AUTHORIZING
      PHONE
      SHIP-TO-ADDRESS
      BILL-TO-ADDRESS
   PART-DETAILS
      PART-NUMBER
      QUANTITY
      SUPPLIER
      PRICE
```

Immediate Accesses

Having defined the necessary and sufficient contents for the data store, we now have to investigate the immediate accesses required. An immediate access can be defined as one which is required faster than it is possible to search or sort the entire data store.

For example, since the supplier of each part is held as a data element in the data store, the question "What orders do we have for Supplier X?" can always be answered, given time, by scanning every record. If the question must be answered while someone is on the phone, however, we must provide for some special secondary access path (an index, a logical pointer or the equivalent).

We use an Immediate Access Diagram as shown in Figure 10 to get a consolidated view of the various immediate accesses required by different members of the user community.

The arrows connecting the boxes "Customer-Name," "Part-Number" and "Supplier-Name" to the box representing the data store are not data flow arrows — they merely represent the necessity for immediate access to a record or records given Customer-Name and so on.

When an immediate access diagram has been drawn up with volumes, file sizes, frequency of access and so on, we can use it as a data base/file design planning tool. If providing all the immediate accesses required presents no technical problem and can be done at acceptable cost, so be it.

However, if the cost of providing all the secondary access paths is unacceptable, then something has got to go. The immediate access diagram is then very useful as a tool to get the users to put some valuation on the relative importance of immediate accesses — obviously if it is very valuable to the business to be able to access pending orders by part number and only "nice-to-have" to access by customer name, then in a pinch we shall do away with the Customer-Name access.

Expression of Logic

The last tool of structured analysis we need to consider is for the expression of detailed logic. We have an

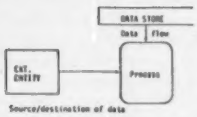


FIGURE 2

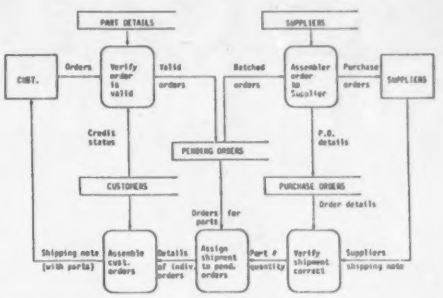


FIGURE 3

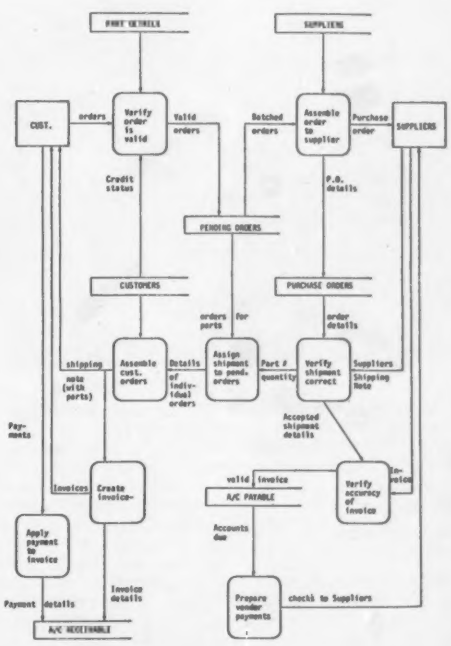


FIGURE 4

overall DFD for the system, with typically 20 to 50 processes in it. The logic of each of these detailed processes should be expressible in one or two pages — but one or two pages of what?

Normally we write logic in natural English. For instance, suppose we exploded "Apply payment to invoice" in Figure 4 and found one of the detailed processes was "Verify discount is correct." Upon inquiring as to the detailed logic of this process, we are given the following memo:

" . . . Trade discount (to car dealers) is 20%. For private customers, 5% discount is allowed on orders for six items or more, 10% on orders for 20 items or more and 15% on orders for 50 or more. Trade orders for 20 items and over receive the 10% discount over and above the trade discount."

This is fairly clear as memos go; however carefully it is written, though, we must read all of it to process any one case. Thus, how long does it take to answer the question, "What discount for a trade customer with a 21-item order?"

We need a more graphic method of displaying the structure of detailed logic, such as a decision tree.

How long does it take you to answer the above question now? The decision tree is an ideal tool for representing the detailed logic of processes which are mainly decisions. In many cases, though, long sequences of actions must be carried out, sometimes repeatedly; then the decision tree cannot be used. We must use our second logic tool, structured English, as shown in Figure 12.

This is a perfectly general, strict dialect of English, using sequence, decision and repetition structures to express the detailed logic of any policy at all.

Figure 13 shows how these various tools fit together to make a logical model either of an existing or a proposed system.

Once the logical model has been developed by the analyst and the user community agrees it meets the system requirements, we can begin system design from a firm foundation. Our first concern is strategic design, sometimes known as "subsystem packaging." What will the subsystems be? What parts of the system function will be automated, and what will remain manual?

Structured Design

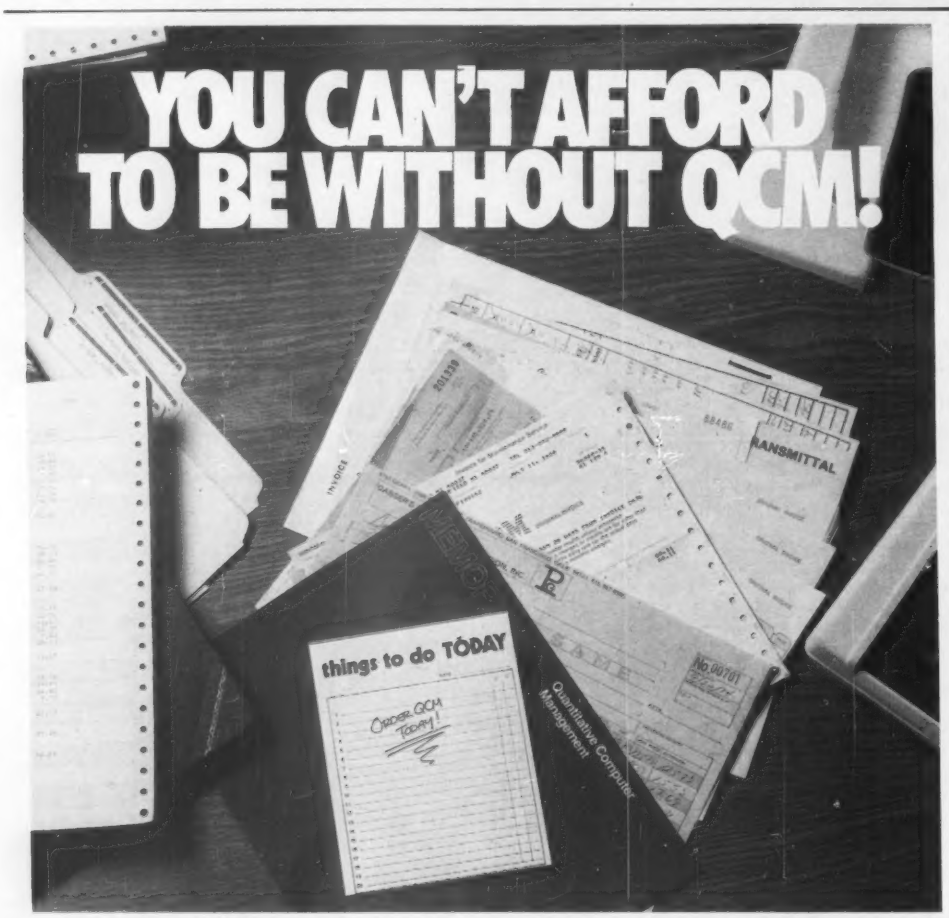
When we have decided on the automation boundary and carved up the computer system into subsystems (creating the minimum number of intermediate files in the process), we have to design the software within each subsystem. Obviously we want a design that works, and we also want a design that is changeable — a modular piece of software whose modules have the least necessary coupling between them to make the system work.

Changeable modular systems frequently look like military command structures, as shown in Figure 14. The system commander gives orders to and receives information from subcommanders, which in turn give orders to worker modules that do the actual reading, writing, editing, computing and so on. Note that workers do not speak to one another — no one speaks unless spoken to, and then only to their immediate superior or subordinate.

This, of course, is exactly what we need to make a changeable system.

By artificially limiting the communication between modules in this way, we are confident that we can change, for example, the logic of the Editor module and, provided the "official" interface between the Editor and the Input Subcommander is not damaged, we know that the changed system will work. In many designs, there are hidden or nonobvious connections between modules such that, if we change one part, then we introduce a bug somewhere else; changing the second part to fix the bug creates a new bug, and so on ad infinitum.

How can we derive such a modular, changeable hierarchy for a subsystem? The most reliable method, initially conceived by Constantine, is to base it on the data flow through the subsystem. (Here, of course, is the link between structured analysis and structured design.)



With today's rising costs, DP managers know they can't afford to be without QCM (Quantitative Computer Management). Your DP dollars must be channeled to those areas yielding the greatest return in service, productivity, and cost savings.

Memorex provides the solution – QCM

Quantitative Computer Management (QCM) is software monitor specifically developed to help you battle the rising costs of computer operations. The information provided by QCM allows you to make intelligent decisions regarding your system's performance. In most cases, these decisions can save you more than the cost of QCM, and do it in the first six months. There are several ways QCM accomplishes these savings:

- Improves your return on computing investment by prompt, concise identification of the most productive use of your current equipment and assuring the greatest benefit from future purchases
- Saves system programmer time by providing accurate, complete and critical data, thus greatly reducing costly guesswork and tedious report interpretation
- Saves user time by fast, precise problem identification and resolution, thereby increasing system availability and performance

See us at Software Info Booth 430

For maximum efficiency and productivity from your current installation, and to ensure intelligent growth, you can't afford to be without QCM. Call or write today for more information on this remarkable system from Memorex for MVS and VSI users.

MEMOREX

Memorex Corporation
San Tomas & Central Expressway
Santa Clara, CA 95052 M/S 12-22
(408) 987-1289

Name _____

Company _____

Address _____

City, State, Zip _____

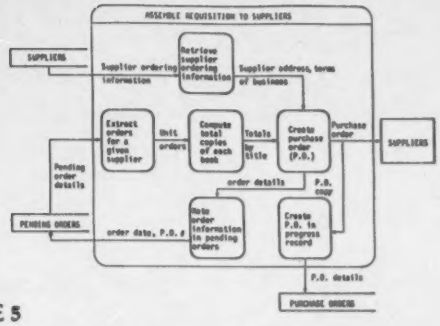


FIGURE 5

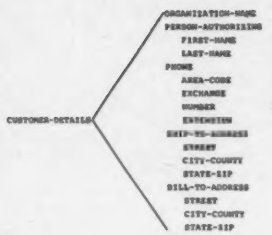


FIGURE 6

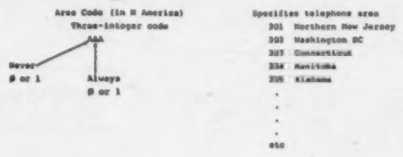


FIGURE 7

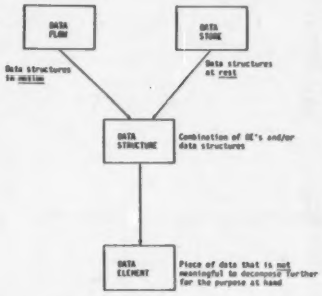

DATA FLOW
DATA STORE
Data structures in motion
Data structures at rest
DATA STRUCTURE
Combination of DE's and/or data structures
DATA ELEMENT
Piece of data that is not meaningful to decompose further for the purpose at hand

FIGURE 8

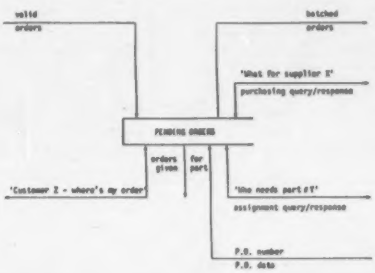


FIGURE 9

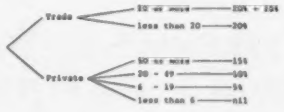


FIGURE 11

```
Add up the total number of items on the order (in the quantity column)

IF    the order is from a trade customer
  and-IF      the order calls for 20 or more
        THEN  discount is 10%
        ELSE  (order is for less than 20)
              NO  discount is 20%
ELSE  (order is from a private customer)
   so-IF      the order calls for 50 or more
              discount is 15%
        ELSE IF the order is for 20 to 49
              discount is 10%
        ELSE IF the order is for 6 to 19
              discount is 5%
        ELSE  (the order is for less than 6)
              NO no discount is given.
```

FIGURE 12

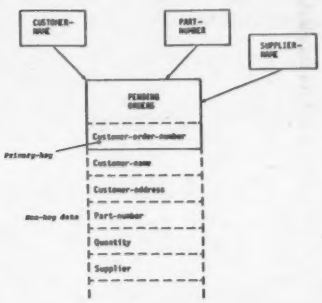


FIGURE 10

Figure 15 shows a very simple detailed DFD with a single stream of data going through it. All such diagrams can be divided into three parts — one part which is concerned with massaging input to get it into a clean processable form (the leftmost two functions in Figure 15); an area which transforms clean input into data ready for output; and an area which massages the output into the required form (the rightmost two functions in Figure 15).

Once we've identified these three areas, we can transform the DFD into a structure chart as in Figure 16, where the large arrows show invocation, the small arrows with white circle tails represent the data passed between modules and the small arrows with black circle tails represent control information such as a flag or a switch.

The structure chart tells us that "Produce Best Solution" invokes "Get Good Input." In turn "Get Good Input" invokes "Read Input," which returns with some Raw Input data.

"Get Good Input" then invokes "Edit Input," passing it the Raw Input data and receiving in return a flag (set to "Good" or "Bad" appropriately).

If the input is good, "Get Good Input" returns Control to "Produce Best Solution" along with the Good Input Data, and so on. Note that each function on the detailed DFD appears on the structure chart as a worker module at the bottom of the hierarchy.

The simple structure chart shown in Figure 16 is a prototype for systems with a single stream of data in the DFD; Constantine called this a "transform-centered" structure. Other structures correspond to more complicated DFDs.

By examining the detailed DFD, we can always use one of these model structures, or a combination of them, to derive a hierarchical design. By marking the data and control interfaces on the structure chart, we can see very clearly on one sheet of paper all the significant aspects of the design, allowing us to review it and see if there is any way it can be made more changeable. This is the type of hierarchical structured design chart to which Carl Garrison referred.

We would not normally expect users to review a structure chart — although, as Garrison said, some may be prepared to do so. Normally, users review DFDs.

User Involvement

We asked Carl what other types of projects on which Superior Oil had used DFDs and structure charts. "We're just getting into design with a geological data base system which captures technical data about geological structures", he replied. "It's a medium-sized project. About half the project team is made up of users — geologists. They've become very enthusiastic about the data flow and the logical model of the system, to such an extent that they're now using it to train geologists who come as new hires!"

McDonnell Douglas Automation (McAuto) has had similar experiences. Jay Green, the section manager responsible for development, said, "I would summarize the benefits of structured techniques as being (a) simplified communication all the way from the final end user through to the programmer, (b) less rework due to the discovery of problems late in the game, (c) more changeable designs, (d) easier estimation of implementation costs, (e) fewer problems found during testing because the structure chart is itself testable."

To what projects has McAuto applied structured techniques? "We have quite a number in process, but we've now completed some projects where we used the structured methodology all the way through. For instance, we did this with our Property Record for Equipment Servicing and Sharing project. This is an IMS on-line project with batch reporting that we brought live in April after 25,000 man-hours of development," Green told us.

Changeable Design

How does the changeability of a structured design work out in practice? Here's Carl Garrison:

"On the revenue accounting project, as we were starting into programming of one section, we were hit with a change in data base requirement. Because of the cut-and-paste nature of a structure chart, making the design change was easy."

How much longer would the same change have taken using conventional flowcharts?

"Very hard to say, but at least five times and maybe 10 times longer."

Words of Caution

We asked Carl and Jay what words of advice they would have for people just starting to use structured techniques.

Jay: "Users must realize that we're going to spend 25% to 50% more time

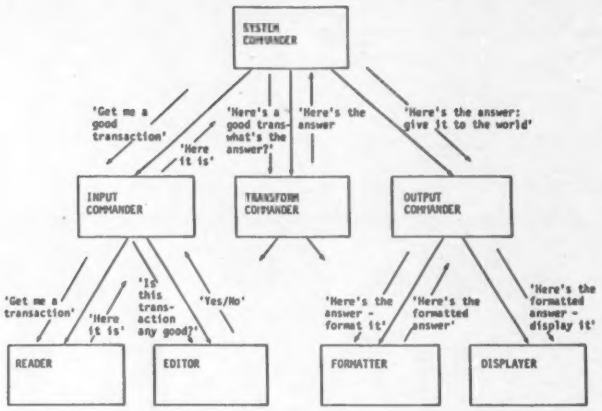


FIGURE 14

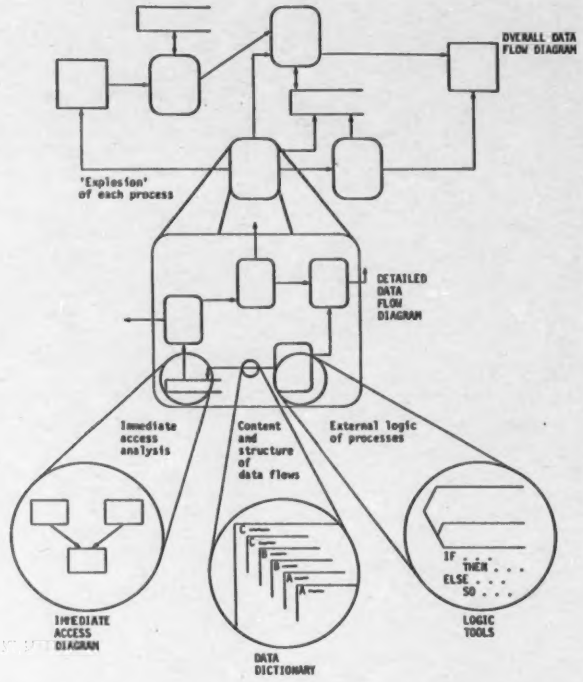


FIGURE 13

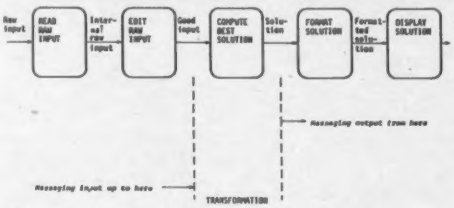


FIGURE 15

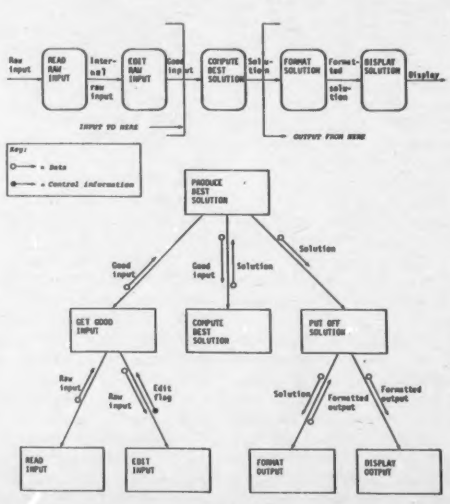


FIGURE 16

SYSTAR HAS THE X.25 BRIDGE FOR YOUR IBM SYSTEM

Systar provides the communications bridge for your IBM system - using the versatile IBM Series/1 hardware. The **CMX Communications Executive** is a terminal cluster controller, an X.25 Host network interface, a message storage and retrieval system, or a communications concentrator.

CMX provides support for multiple terminals and communication lines, a powerful filing system, virtual storage, and a modular programming structure. CMX has protocol support modules for asynchronous, bisynchronous 2780, bisynchronous 3270, SDLC, and X.25 communications. **You can customize the system** for your own distributed processing needs.

Ask us to prove it.

SYSTAR

SYSTAR CORPORATION
1762 Technology Dr., Suite 208
San Jose, CA 95110
(408) 280-7066

and money in analysis and design. Succumbing to user impatience to have programming started before these functions are complete, however, can result in more than compensating time and effort spent in difficult testing."

Carl: "Pick on a project that's large enough to be meaningful, but not so large as to be overwhelming — then use the people from that project as 'seeds' on later projects."

Carl: "It's important to develop procedures ahead of time to tie your DFDs and data structures to a data dictionary."

"We just did a labor distribution system using structured methodology. It wasn't a very large effort, three to four man-months, and it's true the people on it were good, but it was the first project recently to go in ahead of schedule and under budget - and we've had no production bugs in the last three months. A good piece of that is due to 'structured.'"

At Improved System Technologies, Inc., we are pursuing two lines of development for structured methodologies.

The first is the working out of step-by-step procedures and standards to integrate the intelligent use of data flow diagrams, logical data analysis, data dictionaries, structured design, structured walkthroughs and top-down development. This has been done in the structured analysis design and implementation of Information Systems (Stradis), whose procedures and training make up a formal system development methodology.

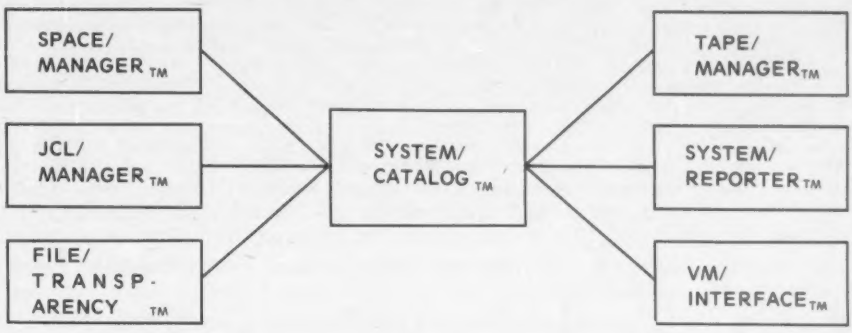
The second line of development is the provision of automated assistance to the analyst/designer. Several groups are experimenting with the use of automated drafting packages to assist in the drawing and maintenance of data flow diagrams, using disk storage and plotter output.

The next step, to our way of thinking, is to build a bridge between the automated data dictionary and the automated drafting package so data and processes defined on a DFD can be transferred to the data dictionary and vice versa.

If we could further integrate the data dictionary with effective text processing, we would have an "analyst's workbench" which would allow the analyst to build the logical model of a system with, say, a dedicated microprocessor, providing on-line entry of data definitions, data flows, information about system objectives, policy statements, etc. It would also permit on-line update and production of DFDs and structure charts (on a plotter) and requirements statements, policy logic, design statements, user procedures, etc., on a text printer in the project area.

While the analyst's workbench would not do any thinking for the analysts, it could speed up the clerical work of analysis and/or enable the rapid sharing of all information between members of the project team.□

FRUSTRATED WITH DOS/VS(E)?



The System/Manager™ is the Solution! ©

SYSTEM/MANAGER is the only fully integrated DOS/VS(E) Resouce Management system with a single catalog and associated utilities for both disk & tape data set management. By automating resource management, SYSTEM/MANAGER allows your operations personnel to run your system at maximum capacity with far fewer errors. By simplifying and expanding DOS/VS(E) JCL, SYSTEM/MANAGER increases programmer/operator productivity. And, as an added bonus, it reduces your disk space and tape requirements. Thus SYSTEM/MANAGER relieves operational frustrations, saves personnel & computer time, reduces costs, and generally improves your D.P. environment. To learn how, call or write us today.

I'd like to learn more -- ☐ Please Call ☐ Please Send Details on how System/Manager can help us.

Name_____
Title _____
Company _____
Street_____
City _____ State _____ Zip _____
Phone: A/C ( ) No. _____
Hardware _____
Operating System_____
S/M CWX

datapro HONOR ROLL

INSAC

INSAC SOFTWARE INC.

1000 Iris Drive
Conyers, Ga. 30207 (404) 922-7235

Security Software?

SAC
The Proven Alternative for Data Security!

Information "leakage" doesn't just happen to the "other guy", it can happen to your company as well. That's why you need the proven protection of the Security Access Controller (SAC) from Electronic Data Systems (EDS), because physical security is not enough. SAC enables you to decide who may gain access to your confidential information, and to what extent. It's the most efficient security software available.

SAC IS UNIQUE . . . a fast, efficient and comprehensive system designed to solve the complete software security needs of a corporation.

SAC IS PROVEN . . . the result of over four years in meeting strict standards for security in a wide variety of business and technical environments. And now available to the data processing marketplace.

SAC IS SUPPORTED by Electronic Data Systems Corporation. One of the largest data processing service companies in the world. And EDS is committed to the continuing support and improvement of SAC.

SAC IS AVAILABLE for immediate use. It can be installed and running within hours. And, EDS is dedicated to maintaining compatibility with current and future releases of IBM's operating systems.

SAC WILL:

- Provide System Entry Protection
- TSO Protection
- IMS-CICS Protection
- Dataset Protection
- Class & Priority Controls
- Adapt to your standards & naming conventions
- Support VS1, MVT, and MVS

SAC WON'T:

- Use Excessive CPU Resources
- Require usage of a valuable technician full-time to maintain
- Require you to write reporting & user exits to meet your accountability requirements
- Require you to redo your security criteria each time your requirements change

Other Products Now Available

WAAPDSUT — The Programmer Productivity Aid
SPLITTER — The SYSOUT Control System
STANFAST —The Standardized Subroutine Package

For more information on how SAC can benefit you call toll-free: 1-800-527-0128 in Texas call: 1-800-442-5791, or write:

Electronic Data Systems Corporation
7171 Forest Lane
Dallas, Texas 75230

EDS

System Software
A tradition of Excellence

COMPUTERWORLD

. . . the one weekly newspaper that's designed to fulfill your unique "need to know" . . . the one newspaper that can give you all the current information you need to benefit you, your organization and your career progress . . . 52 weeks of incisive news, commentary and special reports . . . subscribe today, use the attached postage paid order form.

☐ Please enter my subscription
(details on back)

☐ I'm already a subscriber, but I'd like you to change my:
- ☐ address
- ☐ title
- ☐ industry
- ☐ other

My current mailing label is attached and I've filled in new information on the other side

Put old label or label information here

Please send me COMPUTERWORLD for one year. RATES: United States—$30 Canada, Central & South America—$50
Europe and Near East—$90 Other Foreign—$150 (Airmail Service)

☐ Check Enclosed
☐ Am Ex ☐ BA/Visa ☐ MC (MC O. y - List four digits above your name)

If charge we must have cardholder's signature: _____ Expiration Date: _____

First Initial		Middle Initial		Surname	
Your Title					
Company Name					
Mail To:					
City			State	Zip Code	

Address shown is:
☐ Business
☐ Home

☐ Check here if you do not wish to receive promotional mail from *Computerworld.*

PLEASE CIRCLE 1 NUMBER IN EACH CATEGORY

BUSINESS/INDUSTRY
10 Manufacturer of Computer or DP Hardware/Peripherals
20 Manufacturer (other)
30 DP Service Bureau/Software/Planning/Consulting
40 Public Utility/Communication Systems/Transportation
50 Wholesale/Retail Trade
60 Finance/Insurance/Real Estate
70 Mining/Construction/Petroleum/Refining
75 Business Service (except DP)
80 Education/Medicine/Law
85 Government - Federal/State/Local
90 Printing/Publishing/Other Communication Service
95 Other: _____

TITLE/OCCUPATION/FUNCTION
11 President/Owner/Partner/General Manager
12 VP/Assistant VP
13 Treasurer/Controller/Finance Officer
21 Director/Manager of Operation/Planning/ Administrative Service
22 Director/Manager/Supervisor DP
23 Systems Manager/Systems Analyst
31 Manager/Supervisor Programming
32 Programmer/Methods Analyst
41 Application Engineer
42 Other Engineering
51 Mfg Sales Representative
52 Other Sales/Marketing
60 Consultant
70 Lawyer/Accountant
80 Librarian/Educator/Student
90 Other: _____

COMPUTERWORLD
THE NEWSWEEKLY FOR THE COMPUTER COMMUNITY

CIRCULATION DEPT. 375 Cochituate Road, Rte. 30, Framingham, MA 01701

Detach here, fold, and place in post-paid envelope attached through binding.

By Dixon R. Doll

OPEN SYSTEM ARCHITECTURE & THE COMPUTERIZED CORPORATION

Software is the name of the game for the 1980s.

Ten to 15 years ago, the primary costs in a DP installation were for hardware, with only modest expenditures associated with software and people support. Chances are very good that by the middle 1980s, the costs of software development at all levels — including the people who perform application, maintenance and support, as well as ongoing development — will comprise at least two-thirds to three-fourths of the typical DP budget in major user installations.

Perhaps even more importantly, software holds the answer to some of our nation's most significant productivity problems. One of the major trends of the 1980s is expected to be the significant emphasis by the computer industry in producing new languages and software tools which facilitate the application development process. In so doing, the target is to greatly reduce the manpower time and costs associated with the process of supporting existing applications and developing new ones.

Users in the industry, in general, must get away from continuing to invest in the development of software at all levels which contains either implicit or explicit dependencies on some type of specific computer system, teleprocessing network or physical storage medium associated with the execution of that software. This is what aggravates maintenance costs and frequently causes users to spend more than half of their budget dollars simply maintaining and modifying existing applications. This adversely impacts productivity because it precludes the development of new, high-payoff applications.

The end result of this series of events and problems is continuing inflation, cost overruns in DP budgets and significant problems in the overall application productivity, both in the DP shop itself and in the end-user area.

Enter distributed data processing (DDP). As a result of these difficulties, and the plummeting cost of hardware, end users have begun to take matters into their own hands. The industry is beginning to supply them with DDP configurations that permit them to solve problems without depending on central DP shops.

At the present time, many of the application programs being developed are continuing to be developed in traditional so-called higher level languages like Cobol, PL/I and Fortran. This is not the answer for the future. The real answer lies in the development of specialized languages which will permit application development without programming.

Application development without programming will allow end users to develop solutions to their own problems and to design their man-machine dialogues in a way that closely parallels the actions of the business function. They will be able to design transactions, screen formats for input and report formats for output. They will specify through extremely high-level commands what actions are to be taken. These ultra-high-level languages will then set up all of the appropriate data base interfaces and network interfaces and generate the execution code for the processors which will satisfy the end-user requirements.

In this environment, the need for the DP shop to function as the middle man in the application development process has, at least in theory, been eliminated. Obviously, the DP department will continue to be an extremely important link in the overall chain of events, but its role will change drastically. Satisfying the needs of the thousands of end users around the world generally means increased complexity and sophistication in intermediate (under-the-cover) layers of software, layers which decode the commands of the end user and produce all the proper code that will be executed in the different application processors.

Now let's look at it from a different perspective. All of these higher level application developments without programming concepts demand cost-effective pathways between the terminals and DP machines and data bases, which the end users need to access. The industry is in the preliminary stages of developing a reference model for layered distributed processing architecture as shown in Figure 1 and Figure 2.

The purpose of these layered architectures is to identify logically independent functions and cleanly define interfaces between these different functions which need to go on in an overall DDP configuration.

As the diagrams clearly imply, the first four layers are concerned with the transportation of information from one complex to another. The higher three layers, namely Layers 5, 6 and 7, are concerned with standardizing the protocols and functional activities which go on in the DDP machines at the end points of the transport network. Clearly, software plays a vital role in both groups of layers.

Let us discuss some of the important functions in each of these two categories of layers.

Because the lowest layer, namely the electrical interface, is primarily concerned with the interfacing of a business machine and signal converter, Level 1 is not particularly interesting from a software point of view. It is primarily concerned with the physical properties for the interface.

Nonetheless, software to control the Level 1 interface is important whenever switched connections are involved because the Level 1 interface must be given the dial digits for a circuit switch connection and know how to respond to the various electrical control signals which need to be manipulated at the network interface.

Most of the link level control (Level 2 activity) is concerned with implementing the standard types of data link control that we know as physical link control. Most systems which implement data link control perform it through a mixture of special-purpose software and hardware or through special-purpose microprocessors that are custom-programmed to implement the data link control activity on a dedicated basis.

In general, the microprocessors which perform Level 2 functions are very restricted through the software and hardware, which are implementing these activities. Thus, the software function at Level 2 is to perform and implement the specific protocols utilizing the instruction set of the microprocessor or the special-purpose hardware, which may be designed for those functions as well.

Level 3 software is concerned with the performance of network control. Virtually all Level 3 functions are accomplished through software.

This function is concerned with all the activities that control the usage of the transport network, but not the actual regulation of the flow of data itself. CCITT X.25 protocol, which has already been implemented in many configurations around the world, offers a combined capability for packet switching, network usage and interface that consolidates the functions of Level 1,

Dr. Dixon R. Doll is president of the DMW Group, Inc., a consulting firm in Ann Arbor, Mich.

Control Levels in ISO/ANSI Effort

Level	Example, Scope of Constituent Function
1	Physical Interface between node and signal converter.
2	Physical Link Control
3	Network Control Activities such as addressing, routing. Directing control of switching — not the actual transfer.
4	Transport End-to-End Control provides transparent network interface. Station addressing, flow control and data assurance.
5	Session Control — functions to support setup and use of dialogue.
6	Presentation Control — data codes format information, encryption flags, etc.
7	Process Control — application-dependent functions and conventions.

FIGURE 1

Control Levels in ISO Open System Architecture

FIGURE 2

"A whole new series of communications services, like the plans of Bell's ACS network, will ultimately reside in Layer 4."

Level 2 and Level 3.

Many refinements and improvements to the functionality for the Layer 3 software are still necessary before we have everything that users need. There are many areas where X.25 may be expected to be improved in the future.

These improvements will be accomplished through continual refinements to the standards which permit a full spectrum of priorities; improved sophistication in the routing schemes which can be employed; the automatic establishment of parallel virtual routes through a packet network (to provide a higher capacity end-to-end virtual circuit); improved provisions for the accumulation of traffic statistics and performance measurement; and improvements in the area of using multiple access lines to improve the overall availability of an end-to-end packet-switching network connection.

Layer 4 Software

Layer 4 software is concerned with the specific activities involved in the end-to-end transfer across the network. This so-called transport end-to-end control layer is designed to provide additional functions for the transport network which are not established at Level 3.

For example, Level 3 typically constrains the maximum packet size to a specific length of message. Layer 4 software can provide a mechanism for automatically slicing up an end-user message into chunks that would not exceed these limits.

Other important software functions which will ultimately be provided in Layer 4 involve such activities as a provision of directory assistance functions — the cataloging of data bases that permit users to find out which resources are available on the network. In addition, many enhancements in the network management and administration area will ultimately be provided by the various organizations which provide Layer 4 types of software on their different network configurations.

Layer 4 software will also have to be improved and expanded to permit transport networks which involve the judicious mixture of satellite services, packet-switching services and conventional circuit switching, as well as fast circuit-switching offerings. In addition, a whole new series of communications services, like those unveiled through the preliminary service plans of Bell's ACS network, will ultimately reside in Layer 4.

One could think, then, of software through Layer 4 as the place where the functionality will be provided for the next generation of significant service enhancements to the value-added networks we know today.

Many of these new services will be oriented toward supporting electronic mail activity and the transport of noncoded information, as well as providing user-accessible workspace to facilitate the preparation of messages and provide in-transit storage facilities for bulk data transfers that do not require a physical or virtual circuit setup.

Layers 5, 6 and 7

A comprehensive treatment of Layers 5, 6 and 7 functions would require thousands of pages. It is best to visualize the software functions which support these layers as being all of the typical kinds of activities which the mainframe and minicomputer vendors supply to their customers to support today's teleprocessing and computer network applications.

The general purpose of all of the lower four layers is to permit software that implements Layers 5, 6 and 7 to become standardized and not dependent in any way, shape or form on the specific types of transport mechanism utilized. Vendors want their users to be able to log onto a machine from a remote terminal with exactly the same machine log-on procedure, regardless of whether they are coming in through a local terminal over the public switched network, over a leased line or through a packet-switching carrier. It is important here not to confuse the functions of physical circuit setup and those of session log-on.

When we talk about Layers 5, 6 and 7, we are concerned only with activi-

Using Power/VS(E)? . . . Then You Need SERCON'S®

POWER/VS(E) QUEUE STRIPPER & LOADER

STRIPPER FEATURES

- Up to 750 jobs may be selected from POWER/VS(E). Print, punch & reader queues for output to a Stripper tape or the User Exit.
- Selection by JOBNAME, jobname PREFIX, FORMS, CLASS, PRIORITY, JOB NUMBER, RBM value, DISPOSITION, and SYSID.
- Output tape POWER/VS(E) compatible and may be processed by POWER/VS(E) writers.
- User Exit may be used to format output.
- No modifications are required to POWER/VS(E)
- Selection Report lists jobs selected, their final disposition, and vital statistics.
- KEYWORD=PARAMETER control cards.
- Elimination of multiple tape drives for tape spooling under POWER/VS(E).
- SYSIN tape supplied for easy 3 minute installation.
- Execution from the operator's console or card reader initiation.
- Concurrent operation with POWER/VS(E) or when POWER/VS(E) is inactive.
- Multiple Stripper/Loader jobs may be executing at the same time in different partitions or in different CPUs under POWER/VS(E) shared spool.
- Supports all IBM POWER/VS(E) supported disc devices including FBA technology.
- Transfer of POWER/VS(E) jobs from one CPU another CPU.
- Automatic releasing, deletion, and disposition changing of selected jobs.
- Optional changing of disposition, forms, class, priority, and/or SYSID of selected jobs.
- Optional standard label output for the output POWER/VS(E) tape for users of automatic tape library systems.
- Optional altering of jobs without extraction from the POWER/VS(E) queues.
- Optional listing of the POWER/VS(E) queues without extraction of POWER/VS(E) jobs.
- User specification of Stripper selection default parameters.
- Negative selection criteria allowed (i.e. CLASS #A select jobs not in CLASS A).
- POWER/VS(E) job level enqueuing on selected jobs.
- Support for IBM DOS/VS release 31.0 up.

STRIPPER USES

- Production of host formatted computer output microfilm tapes without modifying application programs.
- Production of offline print tapes.
- Supporting non-standard IBM devices for POWER/VS(E) output.
- Backup of reports.
- Removing POWER/VS(E) "Full Queues" by temporary tape storage.
- Creating punch interpret for jobs in the POWER/VS(E) punch queue.
- Production of POWER/VS(E) tapes for offline microfilm production.
- Producing tapes for the POWER/VS(E) Queue Loader for subsequent loading to POWER/VS(E).
- Backup of large reader jobs in the event of Abends to avoid reloading.
- Backup to RJE Reader jobs in the event of Abends to avoid reloading since RJE station operators may not be available.

LOADER USES

- Retrieving jobs that were backed up by the POWER/VS(E) Queue Stripper for POWER/VS(E) processing.
- Reloading of jobs for multiple transmission to RJE terminals.
- Reloading of jobs that were retrieved for backup or "Full Queue" conditions.
- Transfering POWER/VS(E) jobs from one CPU to another that are not sharing Queues.
- Printing of selected report pages without printing an entire report.
- Loading of POWER/VS(E) output tapes back into the POWER/VS(E) queues.
- Producing Computer Microfilm Tapes from backup tapes for additional microfiche copies.

LOADER FEATURES

- Up to 750 jobs may be selected from tape for loading into the POWER/VS(E) print, punch, & reader queues.
- Selection by JOBNAME, jobname PREFIX, FORMS, CLASS, PRIORITY, JOB NUMBER, RBM value, DISPOSITION, and SYSID.
- Input tape in the POWER/VS(E) format, may be produced by the Stripper, or POWER/VS(E) output facilities.
- User Exit may be used to format output.
- No modifications to POWER/VS(E)
- Selections Report lists jobs selected, their final disposition, and vital statistics.
- KEYWORD=PARAMETER control cards.
- SYSIN tape supplied for easy 3 minute installation.
- Execution from the Operator's console or initiation via the card reader.
- Concurrent execution with POWER/VS(E) or when POWER/VS(E) is inactive.
- Multiple Stripper/Loader jobs may be executing at the same time in different partitions or in different CPUs under POWER/VS(E) shared spool.
- Supports all IBM POWER/VS(E) supported disc devices including FBA technology.
- Transfer POWER/VS(E) jobs from one CPU to another CPU.
- Automatic releasing, holding, placing of jobs in different POWER/VS(E) dispositions.
- User specification of CLASS, FORMS, PRIORITY, DISPOSITION, and SYSID for jobs loaded to POWER/VS(E) queues.
- User specification of Loader selection default parameters.
- Negative selection criteria allowed (i.e. JOB#PAYROLL would select all jobs not having the name PAYROLL)
- Support for IBM DOS/VS release 31.0 up.

- **PRICING (MAINTENANCE INCLUDED)**
  - **One Year Lease-Stripper .....$ 750.00**
  - **One Year Lease-Loader .....$ 750.00**
  - **One Year Lease-Both $1000.00**
- **NO CHARGE ON SECOND CPU SAME SITE**
- **FREE TRIAL - 30 DAYS**
- **OVER 150 USERS**

SERCON corporation

4611 NORTH LINDHURST
DALLAS, TEXAS 75229
(214) 691-3020

"The important need is for balance so that all this cost-effectively distributed processing power will not be used counterproductively."

ties that go on once the transport mechanism has completed the successful establishment of a connection. The software provided by the DP vendors to support the activity of Layers 5, 6 and 7 will be the critical capability which determines their success in the marketplace. Layer 5 software is reasonably straightforward, so most of the action and interest and complexity will be in Layers 6 and 7.

Some of the typical functions which appear to have a high payoff and lots of user appeal in Layer 6 are the so-called virtual terminal and virtual file capabilities. Virtual terminal capability provides complementary software functions at opposite ends of a DDP transport network that permit the preconversion of terminal information from a physical device to a standardized representation for transportation through the network in the virtual terminals' standard format. At the distant end, post-conversion supplies the virtual terminal input to whatever application, program or data base is required.

At the time when the information is presented to the application program or the data base, any physical terminal dependencies have been eliminated as a direct result of the so-called presentation services function in the Layer 6 software.

Virtual file capabilities have the same objective — to remove physical storage media and file accessing dependencies from the application programs, which need to transmit or to access these files through a network. The next logical extension, then, for Layer 6 software enhancement is in the area of distributed data base management systems (DBMS). Clearly, the subject of distributed DBMS is beyond the scope of this article, except to note the importance of that type of capability in the overall scheme of DP vendor and end-user strategies.

Finally, we have the application-level software and supporting protocols. There will be many thousands of types of application-dependent protocols utilized by customers which perform networking. The general objective will be to make it just as easy for an application program running on one machine to utilize all of the power in the layers of software below.

In a broad sense, all of these different layers of software are designed to make it easier for users to solve problems on the ultra-sophisticated distributed DP configurations of the 1980s. As one would expect, the important need is for balance so that all of this processing power, which can now be cost-effectively distributed throughout a corporation's locations, will not be used in a counterproductive way.

Thus, one of the most important elements will be for vendors to provide software which permits the corporation's top management to impose the right kinds of controls over what functional activities can be performed in specific locations and how they will interface and relate to functional business activities at other locations.

In general, then, the resolution of these important control issues and resource allocation decisions for establishing the completely computerized corporation will require an additional complement of software packages which facilitate the design of data, the modeling of data relationships and the modeling of the business planning process itself. Traditional structured programming techniques, as we know them today, will become obsolete. Rather, alert corporations will recognize that application design should be primarily performed to support connections between the users on one side and the data on the other.

Corporations will need to carefully model the end user's requirements for accessing data before they begin application design. Vendors will introduce ultrasophisticated intelligent DBMS that contain embedded logic as well as traditional DBMS functionality. This will have a high payoff because application programs, when developed, will no longer contain the tremendous amounts of functional redundancy which typically exists in the broad portfolio of applications that run in most corporate computer facilities. □

Imagine reviewing simple, easy to understand graphic reports like these instead of volumes of complex computer printouts. Most major oil companies, many leading banks and dozens of other organizations are doing just that, right now, with ISSCO graphics software.
YEARLY EXPENDITURES BY CITY
IMPORTS BY COUNTRY OF ORIGIN
Major U.S. Ports
MAKE BETTER DECISIONS FASTER, STARTING NOW.
THE FASTER YOU CAN REACT TO THE DATA YOUR COMPUTER GENERATES, THE MORE CONTROL YOU CAN HAVE OVER YOUR COMPANY'S PROFITABILITY.
Today, with just a few basic commands, your computer can produce publication quality graphs in a matter of minutes that would have taken an art department days or even weeks to generate.
As the world leader in device independent graphics for over a decade, ISSCO has pioneered practically every major advance in the field. Sophisticated DISSPLA, lets you use the full power of FORTRAN, COBOL, ALGOL and PL/1 to turn theoretical questions into graphic answers. For non-programmers, TELL-A-GRAF, lets anyone use "plain English" commands to produce quality graphs and charts.
The results: reduced costs, increased productivity, plus faster information communication . . . the key to better decision making in the 1980's.
ISSCO
DISSPLA-TELLAGRAF
INTEGRATED SOFTWARE SYSTEMS CORPORATION
4186 Sorrento Valley Blvd.
San Diego, CA 92121
(714) 452-0170 Telex: 697810

TOTAL® AND ENVIRON/1®... ACHIEVED SO MUCH, IT TOOK THE HIGHEST AWARDS TO HONOR THEM.

A Personal Message From Thomas M. Nies
President, Cincom Systems, Inc.

Recently, two Cincom Systems products were honored by International Computer Programs (ICP) with the industry's highest achievement award. These awards recognized TOTAL—our data base management system—and ENVIRON/1—our on-line control system—as the most successful products in their respective categories.

TOTAL is the "first single program product" ever to exceed the $100 million sales mark. By doing so it has become the most successful DBMS—and the most successful software product ever developed.

But, perhaps the success and recognition of our on-line control system ENVIRON/1 is even more remarkable. Because until recently, ENVIRON/1 was *only* available for our TOTAL user community.

Clearly, these awards are an exceptional milestone for Cincom and for the software industry. And in view of the support and acceptance that you, our some 3,000 users worldwide, have given our products over the past decade, we want to send special thanks and congratulations.

Because it has been you—the leaders in all areas of business, including two-thirds of the 'Fortune 100,' —that have helped us evolve a decade of products for all applications and processing environments.

- Your suggestions for new product enhancements and guidance in long-term direction helped us know we were "heading the right way."
- Your hundreds of "references" have helped us help others.
- Your successes have helped make our successes. Because especially today, data processing professionals want to be as sure as possible of data base and on-line success *before* they choose a vendor.

In essence, your exceptional cooperation has helped us develop the foundation data base/data communications software and supporting facilitative software you need for a true data resource management environment. An environment where integrated software tools help generate the quality of information your organization needs.

You have also helped us grow to become the world's leading full-service software supplier. Today, we provide solutions from minis to mainframes *for 28 different systems*, from manufacturing packages to financial packages, from education services to consulting services.

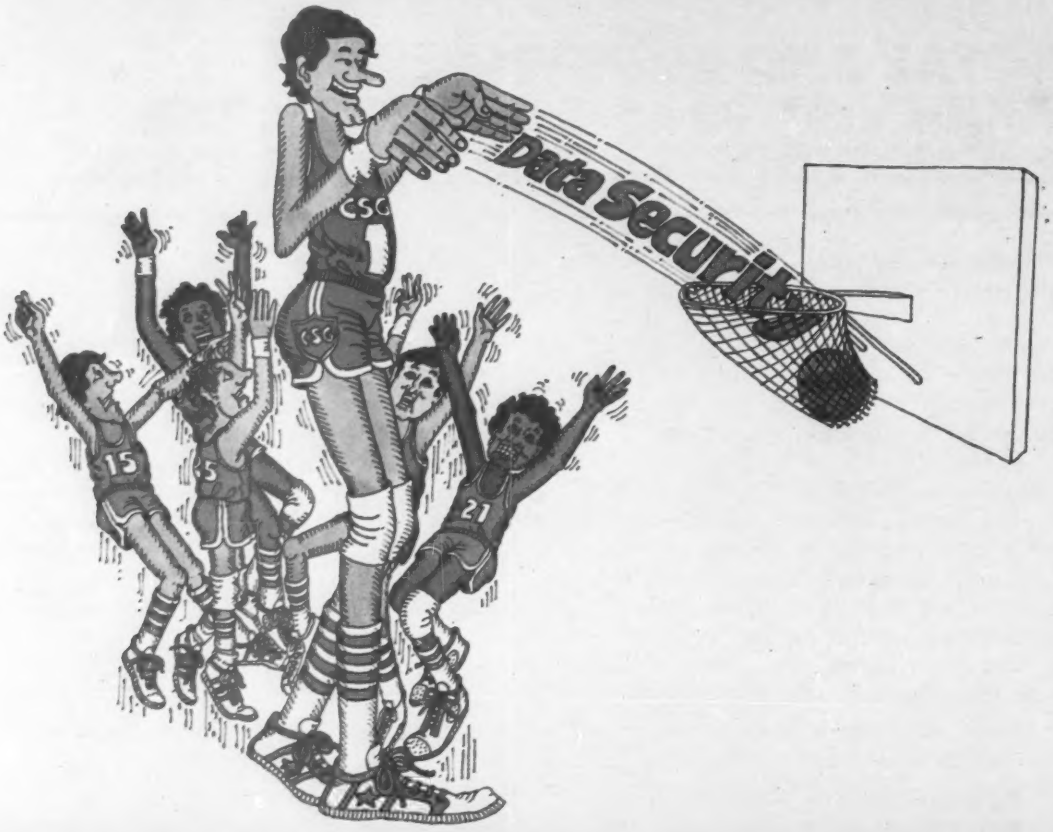
For all this, please accept our sincerest thanks, appreciation—and congratulations. And please accept our pledge to continue doing our very best to support your suggestions and requests in the future.

For you see, Cincom received these awards. But, we know who helped us earn these honors. You did.

Thomas M. Nies
President, Cincom Systems, Inc.

Cincom Systems
2300 Montana Avenue, Cincinnati, Ohio
513 662 2300

ACF2: the Data Security GIANT



IBM may be the giant of the computer industry, but we are unquestionably the **GIANT** in our court.

GIANT concepts: ease of implementation and use! all data protected by default!! centrally or decentrally controlled sharing of data!!! minimum machine and administration overhead!!!!

GIANT appetites: 135 ACF2 sites in its introductory year!!!!! Corporate Contracts with a major car manufacturer and various oil companies; yet also a large number of smaller one-CPU sites.

The Cambridge Systems Group
24275 Elise, Los Altos, CA 94022
(415) 941-4558

Representative: CSG Limited 7 Cavendish Square London W1M 9HA, UK Phone (01) 580-1222 • Telex 299512

Quality products from the Cambridge Systems Group:
ACF2 Data Security. ASM2 DASD Space Management.

ACF2 was developed by Schrager Klemens and Krueger, Inc.

By William B. Engle

OVERVIEW OF SYSTEMS AND UTILITY SOFTWARE PACKAGES

Spurred by the unbundling of the late '60s, the systems and utility segment of the packaged software industry grew tremendously during the '70s. The first part of the decade was marked by a great deal of technical innovation, although it was a period of somewhat uncontrolled growth as the new breed of entrepreneurs probed for their respective niches in this uncharted territory.

Fortunately for the user community, the last several years of the decade brought a much-needed maturity to the industry. Marginally valuable products disappeared from the scene, weak vendors either went out of business or, if they were marketing worthwhile products, were bought out by the more successful vendors. And the dominant vendors honed their products to a high degree of performance and quality. Successes achieved were measured in eight- and nine-figure sales records.

As the decade ended, one product — Cincom Systems, Inc.'s Total data base management system (DBMS) — became the first software package to exceed $100 million in sales, and several others went over the $50 million mark.

Consequently, the '80s have begun with a relatively stable systems software product industry, from which a prospective user can expect to obtain proven, reliable products and effective vendor support. Waiting for the industry to mature or for significant breakthroughs in basic techniques before taking full advantage of the many benefits to be gained through the use of systems software packages is no longer a valid attitude. Considering the tremendous potential for cost-effectiveness in today's systems software, DP budgets in the '80s should include adequate funds for the field, as well as for acquisition of packages for which a payout can be found.

The decade of the '80s will be characterized not so much by dramatic breakthroughs as by the evolutionary development of new products and the progressive enhancement of existing products. As has been the case with hardware developments, these changes will occur within a migratory framework, propelling users along an ever-upward path of enhanced functions.

Integrated Data Management

One of the most visible trends in the evolution of systems software will be the materialization of a concept that is already being billed as "the next generation of software." The essence of this concept is the full integration of data management functions now being performed by separately marketed products that, at best, are only loosely interfaced. This tight functional integration will be accompanied by the restructuring of the data interface to reflect the user's logical view of the data, thereby providing complete user transparency.

Integrated data management software developments have been getting much promotion lately from several leading vendors, namely Cincom, Cullinane Corp. and Applied Data Research, Inc. Others will certainly follow this lead, the most notable probably being IBM, which has made a major corporate commitment to software package development for the '80s.

Implementation of this software development strategy will mean increased ability for users to respond to the ever-greater demands for organization-wide information. Data will become more and more independent of the computer systems and data base structures in which it is processed, and information will be organized and managed in a form that directly reflects the way an organization does business, taking into account the distributed processing needs resulting from geographic dispersion. Information systems will become increasingly user-friendly in every aspect: development, implementation, access, usage, and maintenance. By largely bypassing the traditional channel of DP systems analysis, programming and testing, implementation costs will be greatly reduced.

Even more important, however, the required information will be available within the stipulated time frame, which in tomorrow's fast-paced business world will usually be "immediately." Thus, the integrated software systems now being developed will offer full support to the new corporate information resource management concept that is even now doing much to broaden the traditional DP perspec-

William B. Engle is software editor of Auerbach Computer Technology Reports, *a publication of Auerbach Publishers, Inc., Pennsauken, N.J.*

Following are two listings of suppliers of systems software packages. The first indicates the types of packages offered by major vendors. The second is an alphabetical listing of those vendors, including their addresses and telephone numbers for those who wish to investigate further.

APPLICATION SYSTEM DEVELOPMENT
Applied Data Research, Inc.
CACI
CAP-CPP, Inc.
Decision Strategy Corp.
IBM
Informatics, Inc.
Information Builders, Inc.
M. Bryce and Associates
Programart

DATA BASE MANAGEMENT
Applied Data Research, Inc.
Cincom Systems, Inc.
Computer Corp. of America
Cullinane Corp.
Infodata Systems, Inc.
IBM
International Data Base Systems, Inc.
Intel Corp. (MRI Division)
Mathematica, Inc. Products Group
MSP, Inc.
Software AG of North America
Synergetics Corp.
University Computing Co.

DATA COMMUNICATIONS CONTROL
Altergo Software, Inc.
Applied Data Research, Inc.
Boole & Babbage, Inc.
Cincom Systems, Inc.
Comm-Pro Associates
IBM
Intel Corp. (MRI Division)
Kranzely & Co.
Mathematica, Inc. Products Group
SDA Products, Inc.
Software AG of North America
Turnkey Systems, Inc.
Western Electric Co.
Westinghouse Electric Corp.

COMPUTER GRAPHICS
California Computer Products, Inc. (CalComp)
Decision Graphics
Environmental Systems Research Institute
Escape Ltd.
IBM
Integrated Software Systems Corp.
Laboratory for Computer Graphics and Spatial Analysis
Structural Dynamics Research Corp.
Radian Corp.

DATA CENTER PRODUCTION CONTROL
Johnson Systems, Inc.
Systematics, Inc.
Value Computing, Inc.
University Computing Co.

RESOURCE ACCOUNTING AND PERFORMANCE EVALUATION
Applied Data Research, Inc.
BGS Systems, Inc.
Boeing Computer Services Co.
Boole & Babbage, Inc.
Candle Corp.
Capex Corp.
Computer Associates
Comten, Inc.
Duquesne Systems, Inc.
IBM
Johnson Systems, Inc.
Marino Associates
Performance Systems, Inc.
SDA Products, Inc.
Value Computing, Inc.
Westinghouse Electric Corp.

DASD SPACE MANAGEMENT
Allen Services Corp.
Boole & Babbage, Inc.
Cambridge Systems Group
IBM
Informatics, Inc.
Oxford Software Corp.
Pansophic Systems, Inc.
SDI
Software Module Marketing
Tower Systems, Inc.
Universal Software, Inc.
Value Computing, Inc.
Westinghouse Electric Corp.

PROGRAM DEVELOPMENT AIDS
Adpac Computing Languages Corp.
Applied Data Research, Inc.
Arkay Computer, Inc.
CAP-CPP, Inc.
Capex Corp.
CGA Computer Associates
DASD Corp.
Dataware, Inc.
Hansco Data Processing, Inc.
IBM
Informatics, Inc.
Management and Computer Services, Inc.
Micro Focus Ltd.
MSP, Inc.
Occidental Computer Systems
On-Line Business Systems, Inc.
On-Line Software International
University Computing Co.
Westinghouse Electric Co.

DATA RETRIEVAL, ANALYSIS, REPORTING
Applications Software, Inc.
Azrex, Inc.
Cambridge Computer Associates, Inc.
Capex Corp.
Computer Associates
Cullinane Corp.
Dataman Corp.
Dylakor Software Systems, Inc.
IBM
Informatics, Inc.
McDonnell Douglas Automation Co.
NCI
Optipro, Inc.
Pansophic Systems, Inc.
Program Products, Inc.
SAS Institute, Inc.
SDA Products, Inc.
Software International
SPSS, Inc.
System Support Software, Inc.
Systems Development Corp.

IS&R and TEXT MANAGEMENT
A.R.A.P.
Applied Data Research, Inc.
IBM

LIBRARY MANAGEMENT
Applied Data Research, Inc.
Capex Corp.
Goal Systems
NCI
Pansophic Systems, Inc.
SDI
Tower Systems, Inc.
University Computing Co.
Value Computing, Inc.

OPERATING SYSTEM ENHANCEMENTS
Comten, Inc.
IBM
Innovation Data Processing
Jason Data Services, Inc.
Nixdorf Computer Software Co.
University Computing Co.
Value Computing, Inc.

SOFTWARE VENDORS

A.R.A.P.
50 Washington Road
Princeton, NJ 08540
(609) 452-2950

Adpac Computing Languages Corp.
120 Howard St.
San Francisco, CA 94105
(415) 981-2710

Allen Services Corp.
212 W. National Road
Vandalia, OH 45377
(513) 890-1200

Allergo Software, Inc.
4 Faneuil Hall
Boston, MA 02109
(617) 227-5100

Applications Software, Inc.
21515 Hawthorne Blvd.
Torrance, CA 90503
(213) 540-0111

Applied Data Research, Inc.
Route 206 & Orchard Road
CN-8
Princeton, NJ 08540
(201) 874-9000

Arkay Computer, Inc.
24 Manor House Road
Newton, MA 02159
(617) 964-1894

Azrex, Inc.
3 Mountain Road
Burlington, MA 01803
(617) 272-8750

BGS Systems, Inc.
Box 128
Lincoln, MA 01778
(617) 890-0000

Boeing Computer Services
177 Madison Ave.
Morristown, NJ 07960
(201) 540-7700

Boole & Babbage, Inc.
510 Oakmead Pkwy.
Sunnyvale, CA 94086
(408) 735-9550

CACI, Inc.
12011 San Vicente Blvd.
Los Angeles, CA 90049
(213) 476-8511

California Computer Products, Inc. (Calcomp)
2411 W. La Palma
Anaheim, CA 92801
(714) 821-2011

Cambridge Computer Associates, Inc.
222 Alewife Brook Pkwy.
Cambridge, MA 02138
(617) 868-1111

Cambridge Systems Group
24275 Elise
Los Altos, CA 94022
(415) 941-4558

Candle Corp.
4676 Admiralty Way
Marina Del Rey, CA 90291
(213) 821-2902

CAP-CPP, Inc.
Bldg. 0
1101 State St.
Princeton, NJ 08540
(609) 924-2700

Capex Corp.
P.O. Box 13529
Phoenix, AZ 85002
(602) 264-7241

CGA Software Products Group
25 Commerce Drive
Cranford, NJ 07016
(201) 272-3900

Concom Systems, Inc.
2300 Montana Ave.
Cincinnati, OH 45211
(513) 662-2300

Comm-Pro Associates
638 14 St.
Manhattan Beach, CA 92066
(213) 376-1344

Computer Associates
125 Jericho Tnpk.
Jericho, NY 11753
(516) 333-6700

Computer Corp. of America
575 Technology Sq.
Cambridge, MA 02139
(617) 491-3670

Comten, Inc.
3 Choke Cherry Road
Rockville, MD 20805
(301) 948-8000

Cullinane Corp.
20 William St.
Wellesley, MA 02181
(617) 237-6600

DASD Corp.
9045 N. Deerwood Drive
Milwaukee, WI 53223
(414) 355-3405

Dataman Corp.
700-816 Seventh Ave., S.W.
Calgary, Alberta
T2P 1A1 Canada
(403) 266-4636

Dataware, Inc.
2565 Elmwood Ave.
Buffalo, NY 14217
(716) 876-8722

Decision Graphics
11 Main St.
Southborough, MA 01772
(617) 481-4119

Decision Strategy Corp.
708 Third Ave.
New York, NY 10017
(212) 599-4700

Duquesne Systems, Inc.
2 Allegheny Center
Pittsburgh, PA 15219
(415) 323-2600

Dylakor Software Systems, Inc.
16255 Ventura Blvd.
Encino, CA 91436
(213) 995-0151

Environmental Systems Research Institute
380 New York St.
Redlands, CA 92373
(714) 793-2853

Escape Ltd.
P.O. Box 18797
Atlanta, GA 30067
(404) 255-0404

Goal Systems
P.O. Box 29481
Columbus, OH 43229
(614) 268-1775

Hansco Data Processing, Inc.
Box 236
Wilbraham, MA 01095
(413) 732-9613

IBM
1133 Westchester Ave.
White Plains, NY 10604
(914) 696-1900

Infodata Systems, Inc.
5205 Leesburg Pike
Falls Church, VA 22041
(703) 578-3430

Informatics, Inc.
21050 Vanowen St.
Canoga Park, CA 91304
(213) 887-9121

Information Builders, Inc.
254 W. 31 St.
New York, NY 10001
(212) 736-4433

Innovation Data Processing
970 Clifton Ave.
Clifton, NJ 07013
(201) 777-1940

Integrated Software Systems Corp.
4186 Sorrento Valley Blvd.
San Diego, CA 92121
(714) 452-0170

Intel Corp.
P.O. Box 9968
12657 Research Blvd.
Austin, TX 78766
(512) 258-5171

International Data Base Systems, Inc.
2300 Walnut St.
Philadelphia, PA 19103
(215) 349-9600

Jason Data Services, Inc.
24871 Pylos Way
Mission Viejo, CA 92691

Enter the System 1022® time trials.

Give Your DECsystem A Shot In The Arm.
System 1022 just may be the fastest application development tool you can find – on any system. A data base system designed for DECsystem-10's and 20's, it's received widespread acclaim for speeding up jobs and relieving computer room bottlenecks. An insurance user reports programs commonly completed in 1/10th to 1/50th the usual time. An oil company tells of immediate job turnaround as opposed to weeks. And an engineering company tells us that System 1022 paid for itself on one job alone.

That's Only For Openers.
System 1022 is as appealing to end users as it is to computer mavens. On one hand, non-technical users can learn the system in a day or two, for data entry and retrieval or to generate their own ad hoc reports without programmer assistance. On the other hand, computerniks appreciate its many advanced features such as host language and statistical interfaces, extensive transaction capabilities, audit trails and comprehensive reporting.

So Get In On The Fun. Free.
We'll be happy to give you System 1022 for a free 60-day trial. Try its exceptional performance for yourself. You'll actually be able to do productive work in no time. And you'll find it's fun, if fun means being able to do something faster and more easily than you ever anticipated. Write or call now: Len Goldberg, Software House, 1105 Mass. Ave., Cambridge, Mass. 02138. (617) 661-9440.

The Data Base

Software House

(714) 770-7789

Johnson Systems, Inc.
7923 Jones Branch Drive
McLean, VA 22101
(703) 821-1700

Kranzley & Co.
1010 S. Kings Hwy.
Cherry Hill, NJ 08034
(609) 795-1515

Laboratory for Computer Graphics & Spatial Analysis
520 Gund Hall
Harvard University
Cambridge, MA 02138
(617) 495-2526

M. Bryce & Associates, Inc.
1248 Springfield Pike
Cincinnati, OH 45215
(513) 761-8400

Management & Computer Services, Inc.
Great Valley Corporate Center
Valley Forge, PA 19482
(215) 648-0730

Mathematica Products Group
P.O. Box 2392
Princeton, NJ 08540
(609) 799-2600

McDonnell Douglas Automation Co.
P.O. Box 516
St. Louis, MO 63166
(314) 232-0232

Micro Focus Ltd.
1601 Civic Center Drive
Santa Clara, CA 95050
(408) 496-0176

Marino Associates
7980 Gallows Court
Vienna, VA 22180
(703) 734-9494

MSP, Inc.
21 Worthen Road
Lexington, MA 02173
(617) 861-6130

NCI
3720 Longview Drive
Atlanta, GA 30341
(404) 451-7455

Nixdorf Computer Software Co.
6517 Everglades Drive
Richmond, VA 23225
(804) 276-9200

Occidental Computer Systems
6666 Valjean Ave.
Van Nuys, CA 91406
(213) 782-3005

On-Line Business Systems, Inc.
115 Sansome St.
San Francisco, CA 94104
(415) 391-9555

On-Line Software International
65 Rte 4 East
River Edge, NJ 07661
(201) 488-7770

Optipro, Inc.
P.O. Box 615
Stock Exchange Tower
Montreal, Quebec
H4Z 1J8 Canada

Oxford Software Corp.
174 Boulevard
Hasbrouck Heights, NJ 07604
(201) 288-1515

Pansophic Systems, Inc.
709 Enterprise Drive
Oak Brook, IL 60521
(312) 986-6000

Performance Systems, Inc.
30 Courthouse Sq.
Rockville, MD 20850
(301) 762-0300

Program Products, Inc.
95 Chestnut Ridge Road
Montvale, NJ 07645
(201) 391-9800

Programart
30 Brattle St.
Cambridge, MA 02138
(617) 661-3020

Radian Corp.
8500 Shoal Creek Blvd.
Austin, TX 78766
(512) 454-4797

SAS Institute, Inc.
SAS Circle
Cary, NC 27511
(919) 467-8000

SDA Products, Inc.
475 Park Ave., S.
New York, NY 10016
(212) 481-6800

SDI
P.O. Box 5801
San Mateo, CA 94402
(415) 572-1200

Software AG of North America
11800 Sunrise Valley Drive
Reston, VA 22091
(703) 620-9577

Software International
2 Elm Sq.
Andover, MA 01801
(617) 475-5040

Software Module Marketing
Crocker Bank Bldg./Penthouse
1007 Seventh Ave.
Sacramento, CA 95814
(916) 441-7234

SPSS, Inc.
444 N. Michigan Ave.
Chicago, IL 60611
(312) 329-2400

Structural Dynamics Research Corp.
2000 Eastman Drive
Milford, OH 45150
(513) 576-2400

Synergetics Corp.
1 De Angelo Drive
Bedford, MA 01730
(617) 275-0250

System Support Software, Inc.
5230 Springboro Pike
Dayton, OH 45429
(513) 435-9514

Systematics, Inc.
15 Brookfield Circle
Sterling, VA 22170
(703) 430-0550

Systems Development Corp.
2500 Colorado Ave.
Santa Monica, CA 90406
(213) 829-7511

Tower Systems, Inc.
19782 MacArthur Blvd.
Irvine, CA 92715
(714) 752-8263

Turnkey Systems, Inc.
50 Washington St.
Norwalk, CT 06854
(203) 853-2884

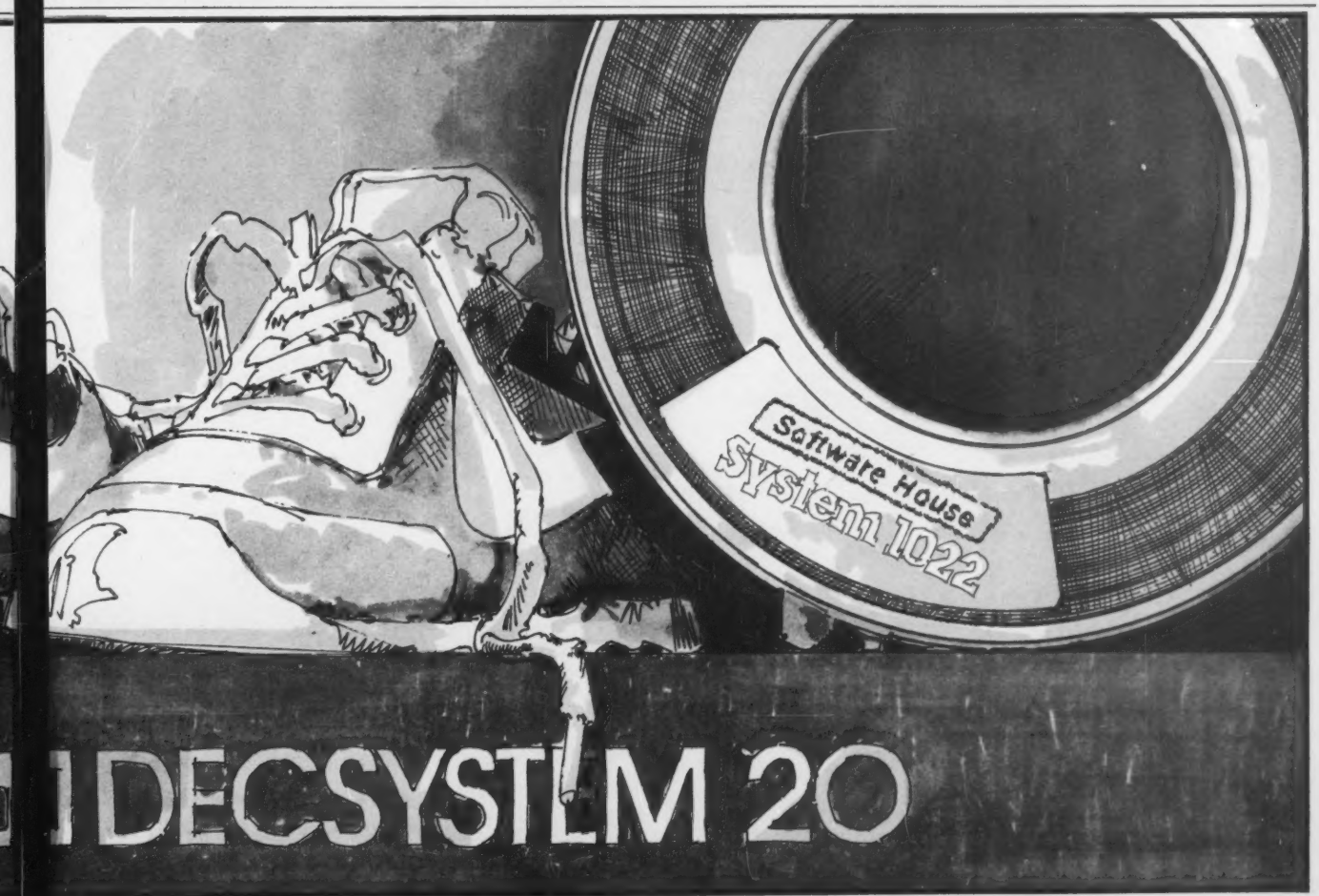
Universal Software, Inc.
Brookfield Office Park
Rte 7
Brookfield, CT 06804
(800) 243-9542

University Computing Co.
Exchange Park
Dallas, TX 75235
(214) 353-7100

Value Computing, Inc.
300 VCI Bldg.
West Marlton Pike
Cherry Hill, NJ 08002
(609) 429-4200

Western Electric Co.
Guilford Center
P.O. Box 25000
Greensboro, NC 27420
(919) 697-2000

Westinghouse Electric Corp.
2040 Ardmore Blvd.
Pittsburgh, PA 15221
(412) 256-5583



tive.

But, as stated earlier, the shape of things to come should not prevent anyone from taking advantage of the many benefits to be reaped by using one or more of today's crop of systems software packages. In this look at the current state of the art, the multiplicity of product offerings has prevented us from treating any of the products, except DBMS, individually. Rather, we have summarized product information by functional category, highlighting the typical features and benefits that can generally be expected of each class of systems software. Background information is included for those who may not be familiar with one or more of the product categories.

A list of major suppliers of systems software packages, indicating the kinds of packages offered, has been included for those who wish to investigate further.

Application Systems Development

It is not without good reason that we begin our discussion of systems software trends for the '80s with discussing the category of systems software most closely associated with applications. The cost of developing applications software is rising almost as rapidly as the cost of the hardware on which it runs is falling. This phenomenon, coupled with the continuing improvements expected in applications software products, will provide the impetus for increasingly greater acceptance of the applications package solution. In fact, it is predicted that the applications segment of the packaged software industry will grow at a significantly greater rate than the systems software segment.

We will not dwell on applications packages except to urge serious consideration of the advantages they offer even when they only meet, say, 80% of prescribed requirements. The trend toward innovative modularization and parameterization of applications packages (under the guise of "application generators") should make it more and more feasible to meet a much higher percentage of application requirements. There will, of course, always be application areas in any organization for which the packaged approach is not feasible. This is particularly true in situations involving on-line, real-time and distributed processing systems. Implementing systems of this kind is a complex and demanding task and, given the rapidly changing nature of today's world, the demands for modification and maintenance of these sophisticated systems will continue. Software that cannot only automate the repetitive aspects of programming, but can make the entire applications development process more streamlined and productive, will be much in demand.

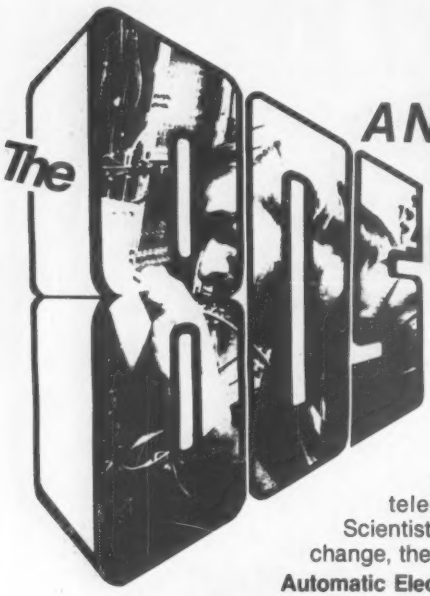
Programmer productivity is not one of the bright spots in DP history, and it continues to be perhaps the most serious stumbling block to the growth of the industry. Programmers — especially good ones — are in short supply and will continue to be so in the foreseeable future.

Although the bidding up of salaries caused by this shortage increases programming costs, the impact on programmer productivity (and availability) resulting from the need to generate more complex programs for increasingly sophisticated applications must also be reckoned with. But, even for those sites that are well staffed with the necessary technical expertise and willing to bear the associated costs, and despite the many excellent program development aids available (which we will discuss later), the time and inconvenience of relying on high-specialty programmers for every new development, every change in a transaction description or screen format represents a major drawback in systems development.

Multitude of Functions

A quick look at the multitude of functions that must be programmed by specialists in developing and maintaining transaction-oriented, on-line applications systems should suffice to point up the need for help.

First, programs must be written for



The

A New Decade...
A New Generation...

BE PART OF IT...
AT GTE AUTOMATIC
ELECTRIC LABS!

We can count on the 1980's to bring exciting and rapid change . . . especially in telecommunications. And for Computer Scientists and Engineers who want to be part of that change, there's no more exciting place to be than **GTE Automatic Electric Laboratories** in Phoenix.

At **GTE,** our engineers utilize the newest generation hardware and software technology. From our on-line, real-time control programs using the latest in structured programming, off-line support, and diagnostic software to our hardware support including IBM 3032/3033 systems, our environment offers a never ending source of professional challenge.

If you have a degree and experience in Computer Science, Electrical Engineering or related discipline, you owe it to yourself to find out more about **GTE**. Currently, we have openings in the following areas:

- **OPERATING SYSTEM SOFTWARE**
- **ON-LINE RECOVERY AND DIAGNOSTIC SOFTWARE**
- **TEST UTILITY SOFTWARE**
- **DEVELOPMENT SUPPORT SOFTWARE**
- **CALL PROCESSING AND ADMINISTRATIVE SOFTWARE**

As a leader in the design and development of advanced digital systems, **GTE** can offer you an excellent salary, complete corporate benefits package and a rare potential for personal and professional development. To be a part of it, send your resume to: **GTE Automatic Electric Laboratories, Manager of Employment, Department CW917, 11226 N. 23 Avenue, Phoenix, AZ 85029.**

GTE | Research and Development

We are an equal opportunity employer m/f/h

"DP personnel are often reluctant to accept the idea of end-user autonomy, but tools to make it feasible are here today."

sign-on and sign-off, security code checking, generation of application screens within transactions, sequencing of transactions and the management of applications module linkage. Programming consideration must then be given to the transfer of data input from screen to screen, as well as from screens to application module and vice versa.

In addition, provisions must be made for handling the intermediate data generated, for posting data to a data collection file, for recovery in the event of a system failure, for reading and reformatting data collection files and for posting data to master files. Finally, adequate documentation and efficient maintenance procedures must be developed.

Development Packages

Systems development software packages are available to perform some or all of these functions without relying on conventional programming techniques. At the very least, these well-developed packages can minimize the need for specialized data communications expertise. The applications specialists themselves, therefore, can handle most data communications aspects of systems development transparently, without additional effort or specialized technical knowledge.

Other significant benefits can be gained, however, especially through the use of the more sophisticated systems development packages. For example, portability of application systems over a wide range of computer equipment and terminal types can be realized, and prototype application systems can readily be developed for the user (or by the user) for evaluation and modification before finalizing the specifications.

DP Reluctance

Another important advantage to an organization that uses systems development packages is user-oriented management of the entire transaction environment. DP personnel are often reluctant to accept the idea of end-user autonomy in the development and management of application systems, but the tools to make this concept feasible are here today; they undoubtedly will continue to be refined and made more adaptable to end-user requirements.

Vendors are accelerating this trend in their marketing efforts by selling their applications and systems development packages directly to end-users. The potential for considerable savings in both time and money guarantees a strong economic impetus to move in this direction.

To summarize, the technology and methodology are available today, in the form of software packages, to allow a user to make virtually a quantum leap toward increasing programmer productivity, improving applications reliability and shortening development and implementation time. Some progress has

There's never been a time when financial control meant more to the success of your company.

And there's never been a financial management system that gives you more control than The Portfolio, from UCS.

Because The Portfolio is the most complete financial management software system on the market today. It's a portfolio of fully integrated systems designed to give you complete financial planning, analysis, reporting and control.

The Portfolio includes accounting systems, a corporate tax compliance and planning system, a planning and forecasting system, a sophisticated business graphics system and reporting module. They are designed to share information from all areas of your company. Automatically.

This means more and better control of your company's cashflow, assets, taxes, budgeting, planning and decision-making in every area and at every management level. So your company won't just survive the rough times ahead. You'll have the control to actually prosper in spite of them.

Learn all the ways The Portfolio can help your company do business more efficiently and more profitably. Write for our free brochure: United Computing Systems, Inc., Business Information Products Division, 2525 Washington, Kansas City, MO. 64108, ATTN: Director of Marketing. Or call (816) 221-9700.

Available on:
IBM OS
IBM DOS
IBM DOS/VSE
Prime
HP 3000
Univac 1100
Univac 9000 series

United Computing Systems

A United Telecom Company
People, systems and solutions United

"There is no substitute for the laborious process of narrowing the choice to the one DBMS best suited to the circumstances."

already been made toward developing truly programmerless systems, and it is predicted that this progress will continue unabated during the '80s.

Data Base Management

The current state of DBMS is characterized by rapidly improving technology at the logical, physical and architectural levels, as well as by rapidly expanding user acceptance. This increased acceptance has come about, at least in part, because of the growing availability of low-cost, direct-access mass storage.

Although a good deal of adverse comment has appeared in the press during the past several years citing the many drawbacks of DBMS implementation, sales of DBMS software continues to set records. According to a recent survey, about half the medium- and large-scale computer systems now being procured involve data base applications. The vast market for packaged data base management software has spurred research and development to provide increasingly more effective products.

The general adaptability of DBMS software is notably difficult to assess and comparative evaulations that do not consider a user's environment are essentially meaningless. There is no effective substitute for the laborious process of narrowing the choice to the one DBMS package best suited to the individual user circumstances.

The following product overviews will serve to illustrate the differences between the leading DBMS packages on the market today. Investigation, however, should cover the entire field and not be limited to the few that are listed here.

ADABAS (from Software AG of North America):

- Employs fully inverted files in which data is retrieved by way of a multilevel index. A single file can be coupled to 80 other files, with up to five coupled files referenced by a single query.
- Permits building of network structures through specification of relationships among records.
- Provides access security down to the field level, with the additional protection of ciphering.
- Stores data in compressed form.
- Operates in serial and multiprogramming environments involving local and remote operations.
- Provides nonprogrammer interface through an easily learned query language (Adascript+) and a report writer (Adacom).
- Includes full integration with Software AG's teleprocessing system (Com-Plete); operates with most other data communications monitors.

DATACOM/DB (from Applied Data Research, Inc. [ADR] Data Communications Division):

- Uses fully inverted file structure.
- Accesses records by way of a multilevel index.
- Permits records to be ordered on a native key for high-speed, sequential batch processing.
- Interfaces with a comprehensive data dictionary facility.
- Supports CALL statements from other languages.
- Permits nonprogrammer interface through English-like, nonprocedural Dataquery language.
- Stores data in compressed form.
- Includes full integration with ADR's Datacom/DC and operates with most other data communications monitors. An Isam interface simplifies applications conversion.
- Runs on IBM mainframes.

IDMS (from Cullinane Corp.):

- Implements most Codasyl recommendations.
- Uses relative chain addressing to relate records in network and hierarchical fashion.
- Coordinates with a comprehensive data dictionary facility (IDD).
- Provides data manipulation language for most high-level languages.
- Permits nonprogrammer interface through Cullinane's optional On-Line Query System and Culprit report generator.

Bobby Harper, Manager of Information Services, Vought Corporation

Vought Corporation makes planning fly with IFPS.®

Vought Corporation, aerospace subsidiary of The LTV Corporation, is a half-billion dollar company which builds military aircraft, missiles and space systems and commercial aircraft components.

Bobby Harper, Vought's Manager of Information Services, recently discussed their use of EXECUCOM's Interactive Financial Planning System (IFPS).

"IFPS was chosen by Vought Corporation in 1976 as our first venture into the area of Decision Support Systems. Comparative studies of products existing at that time proved IFPS to be the only one with a current set of superior capabilities plus unlimited expansion possibilities," he said.

"IFPS has proven itself in direct support of management by providing timely response to management questions that otherwise would have remained unanswered in the time frame allowed. Applications have been most prominent in the financial areas. However, many significant uses are found in manufacturing, materials, and engineering," he noted.

Uses at Vought include debt planning, debt management, treasury operations, consolidation of balance sheet and income statements, transportation, facilities, and engineering planning and forecasting.

He continued, "Answers to critical questions that once were not available may now be researched in a reasonable time. Analysis of data that required production programs and unacceptable lead time to compose routines may now be obtained in time for decision-making department heads to use it."

The challenge of the 80's requires management support that IFPS now furnishes to over 500 users worldwide. Can you afford to wait?

IFPS runs on IBM under OS and DOS/VS, CDC, Univac, Honeywell, Burroughs, Prime, DEC, Harris, and H-P hardware, and is backed by the best professionals in the business.

Contact EXECUCOM today to learn how you can take advantage of the finest available decision support tools with a minimum initial commitment.

EXECUCOM SYSTEMS CORPORATION
Austin, Texas 78766

IFPS®

P.O. Box 9758, Department D3
(512) 345-6560

Sales offices in: Houston, (713) 358-2918; Denver, (303) 494-1222; Chicago, (312) 951-6202; Toronto, (416) 492-7860; New York, (203) 255-2695; Boston, (603) 882-8560; and Washington, D.C.

"Our NCR operating software makes our programmers more effective."

**Richard Shurley, Capital City Press,
Publishers of STATE-TIMES and MORNING ADVOCATE**

SHURLEY:
We've come a long way in a short time. With NCR's advanced software we've gone from a simple batch system to an online virtual memory system that gives us direct access to the data base. In less than two years. We now have the system we need to respond quickly to the needs of our management.

NCR's MILLER:
You were one of our software pioneers, too.

SHURLEY:
We were NCR's 11th VRX site. And the first to install VRX TOTAL, VRX TRAN-PRO and VRX TRAN-QUEST.

NCR's MILLER:
It's very unusual for one user to step out front so often.

SHURLEY:
We kept coming back for just one reason: it kept paying off. It's hard for anyone with data processing experience to believe what we have accomplished with a small, efficient staff and NCR's versatile basic software. Our programmers have really broadened their potential.

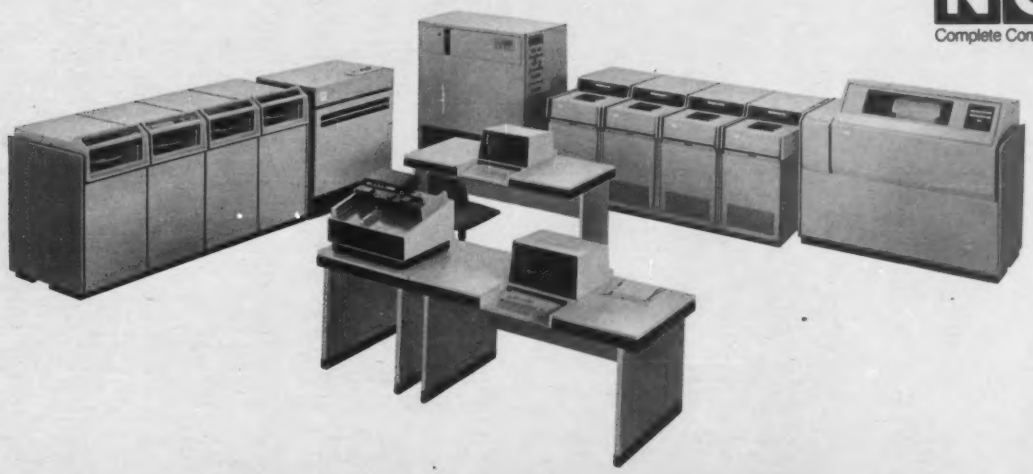
Richard Shurley (right) is Data Processing Manager of Capital City Press, Baton Rouge, Louisiana. Thomas Miller is the NCR representative.

* * *

VRX (for Virtual Resource Executive) is NCR's most sophisticated operating system. TOTAL is the data base management system. TRAN-PRO simplifies transaction processing. And communications programming. TRAN-QUEST makes it easy for anyone to get exactly what he needs to know out of the data base. But only if he is authorized to pass through the security barrier.

In inflationary times, programming efficiency is a particularly important objective. To find out how NCR can help you keep your costs down, phone your local NCR representative. Or write to EDP Systems, NCR Corporation, Box 606, Dayton, Ohio 45401.

NCR
Complete Computer Systems



An NCR computer can make it happen for you, too.

There are still only two ways to give users the online application they want.

The tough way.

And the ACT/1 way.

ACT/1 actually lets you design, and demonstrate new online applications in hours or days instead of weeks or months—and deliver the kind of results users want.

So if you're still using traditional methods to create new online applications, you're doing things the hard way.

The slow, agonizing costly way.

With ACT/1, you interactively create the entire scenario, including screens, dialogue flow and field names with full editing. Management and users see the entire application as you design it, and the way it will be used in actual operation. You can also add your own documentation, and then get hard copy printouts of everything you've created.

With ACT/1, management and users can give approvals or make changes as you go along, often during the scenario demonstration itself. And that's a lot better than the tough way—having to go back to the drawing board weeks or months later and starting over.

Even a novice application developer can learn to use ACT/1 in a few hours. No time consuming coding or special programming is needed to design an application. No in-depth knowledge of CICS/VS, TSO, or INTERCOMM is required. In fact, once a user verifies that your scenario demonstration is exactly what he wants, all that's needed is to add small modules of application code to the dialogue framework—and suddenly your demonstration becomes the actual application.

ACT/1 is one application generator with true multi-system portability. The same ACT/1 application can be transferred between various operating system environments such as DOS/VS/VSE, or MVS/CICS/VS, TSO, INTERCOMM or VM/CMS.

And with all of this, you get far more than an application generator when you buy ACT/1. Because it's part of a growing product line produced by Art Benjamin Associates—a multi-disciplined company with over 60 specialists in the consulting, education, and software product development fields.

The ACT/1 Design Aid, that lets you design new applications interactively sells for only $6,900 to $9,800, depending on environment. The ACT/1 Production System is just $16,100 to $40,000.

But in fact, ACT/1 is such a dramatic concept, one we're so convinced you'll keep, that we'll let you try it for 45 days for just $95—which is our own cost for reproduction, handling and mailing. For that price, you'll get the software, the ACT/1 Developer's Guide, an Installation Guide, a price list, and an Illustrated Guide to the Use of ACT/1.

At the end of that time, just keep ACT/1 and send us any balance, or simply return the package and erase ACT/1 from your system.

There's absolutely no obligation. Except to yourself, when you start doing things the ACT/1 way.

Instead of the tough way.

Art Benjamin Associates Limited
250 Consumers Road,
Willowdale, Ontario, Canada M2J 4V6

Sirs: Enclosed is $95. Please forward the ACT/1 Productivity Package for a 45 day trial period. We understand we can return all materials after 45 days with no further obligation other than to erase ACT/1 from our system. If we decide to keep it, we will forward the balance of the price as printed in the price list received.

Please indicate environment required:

- ☐ TSO
- ☐ OS/CICS
- ☐ DOS/CICS
- ☐ INTERCOMM
- ☐ VM/CMS
- ☐ DESIGN AID
- ☐ PRODUCTION SYSTEM
  - ☐ COBOL
  - ☐ PL/I

Name _____

Company _____

City _____

State/Prov. _____

Zip/P. Code _____

ACT/1 is designed to operate with a system 370 architecture instruction set, VSAM and IBM 3270 display terminal, or equivalent.

Art Benjamin Associates Limited

250 Consumers Road, Willowdale, Ontario, Canada M2J 4V6 · (416) 494-9570

"IMS has a wide range of options that demand a high level of expertise in the design of data base structures."

• Provides fast sequential processing.
• Offers improved system throughput, including multithread operation, through an optional "central version" system architecture.
• Provides compatibility with any telecommunications monitor through a universal communications facility (UCF) option.
• Offers an optional data communications monitor (IDMS-DC).
• Offers an optional DL/1 interface (Escape).
• Runs on IBM mainframes.

IMS (from IBM): IMS provides generalized data base management support for direct-access data sets and is characterized by a wide range of options that demand a high level of expertise in the design of data base structures and file organizations. IMS file structures are basically hierarchies of data segments. IMS features include:
• Data organization methods for the creation, interrelation and maintenance of large data bases designed for multiapplication usage.
• Facilities for the development and maintenance of a data base in a batch environment, with the ability to easily extend data base processing to the telecommunications environment.
• A telecommunications subsystem to support the development of high-volume or rapid response on-line applications systems.
• Remote data processing center interconnection for data base and telecommunications applications.
• Program independence. IMS functions entirely on the basis of subroutine calls from applications programs, which can be written in Cobol, PL/I or assembler.

IMS is available in two versions: IMS-2 (the original product, formerly known as IMS/360) and an advanced version, IMS/VS. Both are available in two configurations, data base only (DB) and data base with data communications (DB/DC). DB configurations can optionally be tied into data communications by use of the IBM Customer Information Control System (CICS).

Two major extra-cost options are provided for the IMS/VS DB/DC system. The Fast Path Facility offers two simplified data base configurations and expedites the processing of high-transaction volumes. The Multiple Systems Coupling option extends message-routing facilities to include transaction input from other IMS/VS systems.

INQUIRE (from Infodata Systems, Inc.):
• Data accessed by multikey indexing in conjunction with field comparison operations.
• Data structures created dynamically when data common to two or more data bases is designated.
• A comprehensive data dictionary facility.
• A procedure language interface for Cobol, PL/I, Fortran or assembler.
• A self-contained, free-format user language for nonprogrammer interface.
• Search strategy implementation using Boolean combinations of keys, field value tests or range criteria.
• A macro feature that provides such functions as prestored routine commands and customized interactive prompting sequences with user-entered variables.
• Word/text processing capabilities, including text search and retrieval, thesaurus browsing and indexing control over single or multiple data bases.
• Backup, recovery and security at the file, record, field or command level.
• Multithreading and protection against simultaneous updates.
• Interfaces with CMS, TSO, CICS and Intercomm.
• Runs on IBM mainframes.

SYSTEM 2000/80 (from Intel Corp), formerly from MRI Systems Corp.:
• Uses a multiple-indexed (inverted), hierarchical data structure for fast retrievals and updates.
• Permits definition of complex network structures.
• Includes a self-contained data base

American Software's Jack Gable says:

"OBS WYLBUR is a classic. It cut costs and improved programmer productivity."

"WYLBUR measured best"
...says Jack Gable, VP of Software Development for American Software Inc., Atlanta.

When ASI had productivity problems, he evaluated the most widely used packages. He selected OBS WYLBUR because it met all his criteria. His programming staff was happy with his choice.

"User acceptance was overwhelming"
...he reports. WYLBUR boosted programmer productivity 40% after only two months. After eighteen months throughput increased 2 to 3 times. It offers the flexibility and ease of operation that your staff will like, too, because it's the multi-purpose, on-line system that enables them to customize programs quickly and efficiently, while eliminating time-consuming, repetitive routine work. You can save time and money, because your programmers will produce more work with far less effort.

"WYLBUR has grown with us"
...he added. ASI found WYLBUR's capabilities more powerful than they anticipated. It's the all-in-one system that grows as you grow, expanding to meet your needs. You can streamline your operation with WYLBUR, and significantly add to your scope...all for very little cost.

"Outstanding response time"
...said Gable. WYLBUR is designed to run at maximum efficiency, providing minimal impact on batch throughput. WYLBUR's 1 to 3 second response time was 5 times faster than competitive systems.

"We have experienced highly professional and unusual personal service"
...concludes Jack Gable in summing up his relationship with On-Line Business Systems, Inc.

Let OBS show you a better WYLBUR that gives you "big shop" capability without "big shop" costs. Through increased programming productivity, you can expect OBS WYLBUR to pay for itself in six months or less.

Take advantage of our **free trial offer.**

For immediate information, phone (415) 391-9555 or simply fill out the coupon below.

ON-LINE BUSINESS SYSTEMS, INC.
115 Sansome Street San Francisco, California 94104 (415) 391-9555

_____ I'm interested in a free trial of the better WYLBUR by OBS
_____ Send more information

NAME_____TITLE_____
COMPANY_____PHONE_____
ADDRESS_____
CITY_____STATE_____ZIP_____
OPERATING SYSTEM_____

*WYLBUR is a registered trademark of Stanford University

definition, query/update and administration facility that uses English-like vocabulary and syntax.

- Uses a "where" clause to support a wide range of subset processing and Boolean logic to expedite ad hoc queries.
- Includes a program language extension for interfacing with Cobol, Fortran, PL/I and assembler.
- Provides data independence and security at the field level.
- Includes integrated data dictionary.
- Features a modular design that permits selective acquisition and upward mobility.
- Includes full integration with MRI's TP 2000 and interfaces with TSO, CICS and Intercomm.
- Stores data in compressed form.
- Runs on IBM mainframes, the Control Data Corp. 6000/Cyber and Univac 1100.

TOTAL (from Cincom Systems, Inc.):

- Features the choice of a randomly organized or a bidirectional, multilist (partially inverted) file structure, either of which can be defined as a logical network.
- Permits the creation, maintenance and retrieval of data in both batch and on-line modes.
- Includes a nonprocedural data management language that interfaces with any host language having a CALL or EXIT facility.
- Achieves data independence and security through field-level addressing.
- Handles physical disconnections and resets pointers automatically.
- Provides a nonprogrammer interface through an integrated nonprocedural language (Socrates) for batch mode and a query facility (T-Ask) for on-line mode.
- Includes full integration with Cincom's Environ/1; operates with most other data communications monitors.
- Includes a comprehensive data dictionary facility.
- Runs on a variety of mainframes and minicomputers.

Some Points to Consider

Every DBMS has certain restrictions on the way in which the designer can structure the data base. These restrictions can be overcome by defining additional record types and relationships; however, this extra structure produces what is known as "clutter."

When a detailed DBMS selection study is performed along with an analysis of the applications data, the clutter that must be introduced into the data base design because of the shortcomings of a particular DBMS will become apparent and can be further evaluated.

Another point to consider carefully is that DBMS vary considerably in the efficiency of their generated code. Even if the data base is reasonably well defined within the confines of the structuring facilities, it is still possible for the user to get a shock when applications program execution is measured. This is particularly true for on-line systems, in which terminal response time is so noticeable.

The basic difficulty with hidden execution time overhead is that the DBMS is designed so that applications programmers can easily write programs that are logically and functionally correct and will produce the desired results. This assumes that the applications programmer need not know what goes on "behind the scenes."

In order to write reasonably efficient code, however, the programmer must have this knowledge. Consequently, an ease-of-use advantage may not truly exist. To circumvent this problem, careful analysis of DBMS structure and code is necessary. In addition, emphasis must be placed on training programmers up to the level of expertise required to work the new system.

Although the achievement of improved data independence is frequently cited as one of the prime reasons for using a DBMS, it should be kept in mind that most systems must be extended after a certain period of use. This extension may involve designing and preparing new programs to process the existing data base.

It also may be necessary to add new items to existing record types, to add completely new record types or to create new relationships between new and existing record types. If this can be done without affecting existing applications programs, the DBMS can be said to support a reasonable level of data independence. With some commercially available DBMS, however, it is simply not possible to add new data

FOLLOW THE LEADER.

DataStar.™ A general purpose key to disk data entry software package that's way ahead of the pack.

DataStar has two distinct phases. One allows you to actually design on the CRT the exact form you need. The other allows you to store and retrieve data. Quickly. And accurately. So, when it comes to data entry and verification, go with the leader.
Go with DataStar.

WordStar.™ The world leader in microcomputer word processing.

With so many advanced features, isn't it quite obvious? WordStar is a born leader. In fact, more than seven thousand people have purchased WordStar from over 300 dealers around the world. In just the first year! Proof positive that WordStar is *the* word processing software package for Z-80, 8085 and 8080 microcomputers.

SuperSort.™ It makes sorting so simple, it's simply in a class by itself.

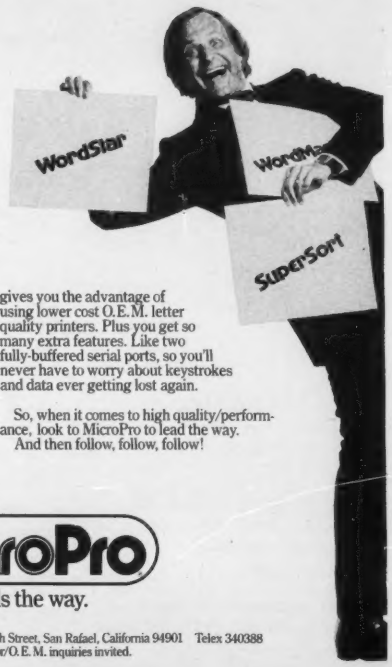
There you are. With a *mountain* of tedious tasks. But don't panic. Here comes SuperSort to the rescue! SuperSort is faster, more powerful and able to sort 560 records in a single minute. Hooray for SuperSort. It makes molehills out of mountains.

WordMaster.™ It's everything you've ever wanted in a video text editor. And more.

WordMaster is so powerful and flexible it can actually double as a word processor. Yet it has all the capabilities you need to enhance your file management needs on a small budget. No wonder it's the path most programmers wish to take.

I/O Master.™ The economic route to quality printing.

Now you can get better input/output performance for less money because I/O Master's S-100 board gives you the advantage of using lower cost O.E.M. letter quality printers. Plus you get so many extra features. Like two fully-buffered serial ports, so you'll never have to worry about keystrokes and data ever getting lost again.



So, when it comes to high quality/performance, look to MicroPro to lead the way.
And then follow, follow, follow!

MicroPro
Leads the way.

MicroPro International Corporation 1299 4th Street, San Rafael, California 94901 Telex 340388
Dealer/Distributor/O.E.M. inquiries invited.

Prices and specifications subject to change without notice.

"It is important to assess data independence functions when performing DBMS comparative evaluations."

items to existing record types.

The solution then is either to add extra record types and relationships, with the concomitant clutter and efficiency problems, or to undertake an extensive rewrite. Because this can be extremely expensive and negates one of the prime reasons for using a DBMS, it is most important to thoroughly assess data independence functions when performing DBMS comparative evaluations.

Packaged Support

Finally, mention should be made of the software packages that support or extend the usefulness of DBMS or that provide utility in their design and maintenance. These packages fall into the following categories:

• *Data Dictionary/Directory Systems (DD/DS).* The DD/DS is the repository for all definitive information about data in an organization: its identifying characteristics, relationships and authorities. The dictionary function informs the user of the entire content of data stored in the organization's data base; the directory function indicates the location of each data element.

Most DBMS either have an integrated DD/DS or can be interfaced with one produced by another vendor. A DD/DS can be quite valuable as a stand-alone system in regimenting an organization's data resources. In fact, installation of a DD/DS prior to DBMS acquisition can smooth the implementation process.

• *Query Languages.* Most DBMS offer their own languages for on-line query of the data base, usually as extra-cost options. A number of independent vendors market query language packages, some offering unique features, that compete effectively with those offered by the DBMS vendors.

• *Data Base Design Aids.* Packages in this category aid in the complex process of definition, design and implementation of a DBMS. They typically provide a development methodology by introducing a terminology patterned on generic data base interactions that are reflected in the data gathering processes. Data base design aids also provide computer-driven analyses with which the data base designer can interact in order to derive an optionally structured data base.

• *Data Base Maintenance Utilities.* A number of vendors have found a fruitful field in supplying innovative packages that assist the data base administrator in maintaining data base integrity. In view of the large number of IMS users and the well-publicized difficulties of maintaining this DBMS, it is not surprising that most of the maintenance packages are designed to operate with IMS.

Typical functions include space management, reporting on pointer discrepancies, assisting in data base reorganization, generating test data bases, reporting on the characteristics of the transaction workload and providing statistics for performance tuning.

Data Communications Control

When the computer is used outside the straightforward batch mode for teleprocessing or data communications, it is necessary to consider software that must perform a variety of complex tasks. Since operating in a data communications mode is rapidly becoming the rule rather than the exception, it would be helpful to review briefly the kinds of tasks that must be performed before describing the characteristics of the software designed to perform them:

• **Communications Tasks.** Communications tasks involve all network-oriented functions not assumed by the network controller or equivalent equipment. These tasks provide the message interface between the network and all the program elements of the system. Communications tasks consist of:

(1) Initiation of Network Activity — This includes polling (inviting terminals to transmit), calling (directing terminals to receive), dialing stations on switched lines, answering incoming calls on switched lines and the like.

(2) Message Assembly and Disassembly — This involves the assembly

Two Distinguished Products for PDP-11... And now VAX Users

INTAC™
MAPS™

Interactive Data Base Management

INTAC is a new concept for data storage and retrieval that features an easy-to-use question and answer format, built-in edit rules, multi-key ISAM data access, interactive inquiry and a unique report generator.

Financial Modeling

MAPS, recognized worldwide for over five years as a leader in financial modeling and reporting, is used to construct budgets, financial forecasts, consolidations and "what if" analyses.

Ross Systems, with over seven years of proven capability, now offers these two products to current and prospective PDP-11 and VAX users. INTAC and MAPS enable business managers to produce instant reports themselves, and relieve DP managers from the pressures of special requests.

Ross Systems offers these management tools on our timesharing service, for license on existing computers and as part of a complete, in-house timesharing installation.

Call us collect for more information.

ross systems incorporated

1900 Embarcadero Road, Suite 208, Palo Alto, CA 94303 • (415) 856-1100 • Other offices in San Francisco and Los Angeles

of incoming data blocks into complete messages and the corresponding disassembly of outgoing messages into the appropriate block structures. The message assembly/disassembly process may require the staging of message data to or from secondary storage, especially in a message-switching environment in which information recoverability is required.

Thus, message assembly/disassembly includes all conversions between standard interval message formats and terminal-oriented message formats.

(3) Protocol Functions — These are procedures required as part of the operating terminal and processor protocols. These procedures include data framing, idle character fill, redundancy checking, block acknowledgment, status reporting and terminal control.

(4) Error Detection and Recovery — These are software tasks required for recognizing and treating network errors. Both line- and status-type errors can occur on either a transient or permanent basis. Transient errors are treated by performing appropriate retry activities. An error that is not corrected after a specified retry threshold is considered permanent and is dealt with by generating appropriate notifications and then usually reducing the level of service to the malfunctioning component.

• **Message Management Tasks.** The message management facilities in a communications system provide the central core of functions that handle messages as logical information units. This core can include the full complement of message processing (as in a simple store-and-forward message switch) or can represent a foundation of services used by user-oriented applications tasks.

In either case, message management usually includes the following functions:

(1) Routing — This involves associating messages with appropriate destinations (e.g., to terminals, other computers or processing programs within the communications system.).

(2) Validation — This involves preventing erroneously formed or addressed messages from entering or passing through the communications network. Erroneous messages can be purged, routed to an intercept terminal for correction or returned to the originator for reentry.

(3) Disposition — This involves removing messages from the residence medium after completion of all required deliveries, thus releasing message space for use by arriving traffic.

(4) Queue Management — This involves maintaining and manipulating message queues within the system, including adding messages to a queue, moving messages from queue to queue and removing messages from queues. Most communications systems use queuing in message management to provide the input, nonsynchronous operation of the message input, message processing and message output tasks.

In addition, priority levels are usually provided with queue structures to ensure proper response to messages requiring rapid action.

• **Error Control Tasks.** The rate of error occurrence in a network, including lines, terminals, modems, switches and other interconnect equipment, tends to be greater than the rates usually experienced in the operation of standard peripheral devices. Consequently, communications operations require more extensive error detection capabilities (to recognize and classify errors as to type and location) and greater variation in retry procedures.

Communications network errors can be classified as follows: data errors that involve a recognized transformation between the information sent and that actually received, and data errors that can be detected by recognizing improper framing, incorrect character parity, receipt of a character illegal in the code set or receipt of a block check character or sequence not identical to that computed for the received data. Nondata errors include all other recognizable malfunctions, such as loss of incoming data carrier, modem, terminal ready status and so on.

Now...

MCBA has DEC Manufacturing Packages!

See Us At SOFTWARE INFO BOOTH #323

In Chicago, September 22-25 at the Hyatt Regency Hotel where we will be demonstrating our Accounting packages for DEC, Data General, Wang, Texas Instruments and Hewlett-Packard mini-computers—and our new DEC Manufacturing Packages.

MCBA is the acknowledged leader in business application packages for mini-computers with an estimated user-base of over 5,000 sites.

Now, after several man-years of development, MCBA is releasing a set of manufacturing packages called MAPs™ (Manufacturing Application Packages) for users of DEC mini-computers.

The following manufacturing packages are available in DIBOL® for use on selected DEC PDP®-11s:

- **Inventory Management**
- **Bill of Material Processor**
- **Customer Order Processing**
- **Sales History**

Also available are:

- **Accounts Receivable with Sales Analysis**
- **Accounts Payable**
- **Payroll**
- **General Ledger**
- **Fixed Assets and Depreciation**

For more information, contact an MCBA Manufacturing Representative at our Glendale, California offices:

MCBA® **Mini-Computer Business Applications, Inc.**
117 South Brand Boulevard
Glendale, California 91204
Telephone: (213) 247-9050 Telex: 194188

Copyright© 1980 by Mini-Computer Business Applications, Inc. MCBA® is a registered trademark of Mini-Computer Business Applications, Inc. DEC® PDP® and DIBOL® are registered trademarks of Digital Equipment Corp.

"The identification and diagnosis of line outages and network malfunctions can be provided by data communications software."

• **Network Supervision Tasks.** Supervisory control of a network is usually implemented by a set of operator commands that allows the selective enabling, disabling and functional modification of the network. Control commands are entered through the system console device or by way of a specially designated set of network supervisor terminals. Each command is acknowledged (either positively or negatively) in order to provide a complete record of operator-initiated action. Commands usually available to the network controller are:

1) Disabling or enabling a line or station for input or output.

2) Alternating route traffic from one station to another and changing the order and/or frequency of line or terminal service (polling, calling, dialing, etc.).

• **Applications Tasks.** The applications tasks in a data communications system handle message processing and other functions not provided by the basic software components previously described in this section. Such processes as text editing, data base inquiry, file updating and collection are applications tasks often incorporated into communications processing systems. These tasks use the facilities of the operating system, communications software and message management to obtain messages for processing and release them for eventual delivery or termination; new messages are often created as part of applications processing.

Typical applications supported by data communications software include:

1) Remote Job Entry (RJE) – This involves the collection, preprocessing, reprocessing and transmission of batches of data from a terminal workstation to a host computer. With the advent of distributed processing, most pre- and post-processing, as well as user access to data files, are performed at the point of origin.

2) Time-Sharing – This is a method of sharing the processing power of computers concurrently. Time-sharing is a terminal-oriented type of multiprogramming designed to facilitate program development, compilation and testing, as well as data base updating and retrieval.

3) Communications Monitoring and Diagnostics – When analyzing the efficiency of a network, its hardware and various supporting programs, values representing the number of characters, messages and acknowledgments/terminations can be recorded and evaluated. In addition, the identification and diagnosis of line outages and network malfunctions can be provided by data communations software.

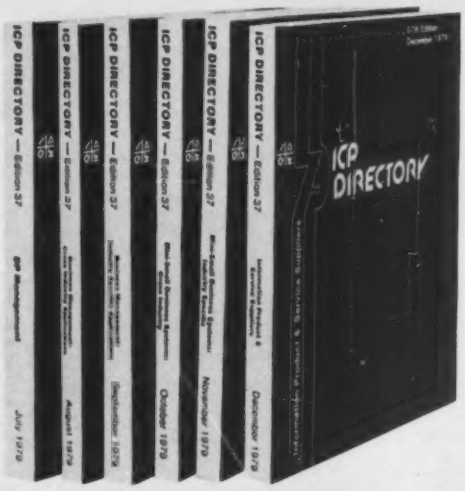
Access Methods

Procurement of software for the performance of the many tasks outlined in the preceeding usually begins by acquiring from the computer vendor a "telecommunications access method," which controls the transmission of data between main memory and the various data communications devices. Since the IBM telecommunications access methods are the most widely known, they are described here to illustrate the types of functions performed by this software:

• The IBM Basic Telecommunications Access Method (Btam) is an elementary program that handles the transmission and receipt of data as well as the polling and addressing of terminals. All other functions must be programmed by the user.

• The IBM Queued Telecommunications Access Method (Qtam) incorporates such additional communications facilities as scheduling and allocating resources, logging message activity and queuing messages on disk. A macro language is provided so network control functions can be performed from within the applications programs.

• The IBM Telecommunications Access Method (Tcam) an enhanced successor to Qtam, includes a message control program for handling message and network control functions and providing facilities for system recovery, checkpoint/restart procedures and control procedures for system shut-



ICP DIRECTORY '80

Information Processing Solutions For The 80's

The *ICP DIRECTORY '80* was first published in 1967 and is recognized as the foremost source for solutions to information processing problems. Over 7000 products and services, and over 2,000 corporate descriptions of IP&S vendors are contained in six volumes, which are updated every six months.

THE ICP DIRECTORY
—Six Volumes—
—Published Twice Each Year—

Data Processing Management — systems products & services for medium & large computers.
Business Management: Cross Industry Applications — applications product & services addressing the administrative, accounting & financial areas of any business discipline.
Business Management: Industry Specific Applications — products & services for medium & large computers developed for use by a specific industry or business discipline; such as, banking, insurance, manufacturing, etc.
Mini-Small Business Systems: Cross Industry — systems & general applications products & services written for mini & micro computers & used by any industry.
Mini-Small Business Systems: Industry Specific — more applications written for specific business disciplines using mini & micro computers.
Information Product & Service Suppliers — Complete corporate descriptions of vendors in the information processing industry.

10-Day Free Trial

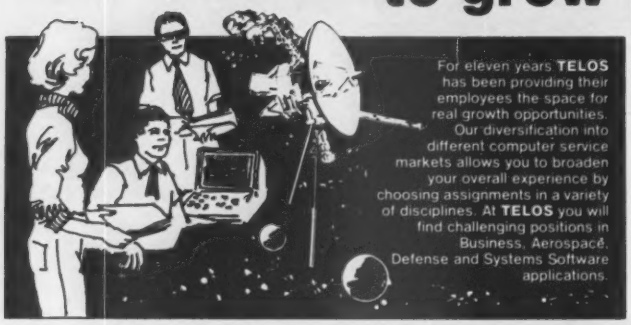
You can have the ICP DIRECTORY '80 for 10 days, with no obligation to purchase. Use it to evaluate information processing solutions by cost, specifications, competition and more.

☐ Send me additional information on the ICP Directory '80.
☐ Please have a Sales representative call me.
☐ Send the ICP Directory '80 on a 10-day trial, with no obligation to purchase.

Name _____ Title _____
Company _____
Address _____
City _____ State _____ Zip _____
Telephone Number _____

If you need immediate information on the DIRECTORY '80 call toll free 1-800-428-6179
International Computer Programs, Inc. • 9000 Keystone Crossing • Indianapolis, Indiana 46240

Space for your career to grow



For eleven years **TELOS** has been providing their employees the space for real growth opportunities. Our diversification into different computer service markets allows you to broaden your overall experience by choosing assignments in a variety of disciplines. At **TELOS** you will find challenging positions in Business, Aerospace, Defense and Systems Software applications.

Currently **TELOS** has immediate openings available for Programmer/Analysts at locations throughout the nation:

CONNECTICUT
Norwalk

CALIFORNIA
Santa Monica - Orange County - Pasadena - Oxnard/Ventura - San Diego Santa Barbara - Sunnyvale

MASSACHUSETTS
Boston

OKLAHOMA
Lawton (Ft. Sill)

VIRGINIA
Hampton

WASHINGTON
Seattle

Equal Opportunity Employer M/F

We seek individuals who share our appreciation for good performance and pride in workmanship and who are in the top 10% of the realtime Systems/Programming community.

If you're serious about a career and want to join a company that offers space for personal and professional advancement coupled with an excellent salary and benefits package - - - contact us today.

Penny Bevier
TELOS COMPUTING, INC.
3420 Ocean Park Blvd.
Santa Monica, California 90405
(213) 450-2424

TELOS

HP-3000 USERS!

• ACCOUNTS PAYABLE • GENERAL LEDGER
• ACCOUNTS RECEIVABLE • JOB COST
• PAYROLL

UTILIZES: • IMAGE DATA BASE
• FORMATTED SCREEN ENTRY

LESS THAN $20K
Our Local Distributors are:

(AZ) Automated Data Systems, Corp. — (602) 244-8222
(CO) Barber, Mabry, Cato & Assoc., Inc. — (303) 771-7734
(CA) Coastal Information Systems — (213) 864-8080
(FL) Foresight, Inc. — (305) 920-9604
(CA) Gentry, Inc. — (415) 547-6134
(NY) Software Systems Technology, Inc. — (212) 964-9600

National Systems & Services
3130 LaSelva, Suite 307
San Mateo, CA 94403
(415) 574-5655

☐ AP ☐ AR ☐ GL ☐ JC ☐ PR

Name _____
Title _____
Firm _____
Addr _____
City _____ St _____ Zip _____
Phone ( ) _____

down.

• The IBM Virtual Telecommunications Access Method (Vtam) was designed specifically for IBM's virtual operating systems (DOS/VS and OS/-VS). Vtam not only has all the facilities of Tcam but operates across multiple partitions and provides dynamic interfacing with applications programs.

Monitor Packages

The second major software item, which offers a selection of vendors and features, is the data communications monitor package. IBM's CICS is undoubtedly the best known package of this type.

Basically the data communications monitor provides the interface with the applications programs and, therefore, with the end user.

A variety of software packages is available for assistance in network planning, design and maintenance. For the CICS user, many additional utilities are offered to help in such system design and programming areas as display management, screen formatting, resource allocation/deallocation, spooling, access control, message transmission and performance enhancement.

Computer Graphics

The old adage that one picture is worth a thousand words is as true today as it was in antiquity, although today the picture should also be related to "numbers." Graphics have long been used for important presentations in which the summarization of a lot of information must be communicated quickly and with strong impact.

However, the time and cost of preparing such material manually is great, and updating it when necessary is not always feasible. The computer is changing all that.

Through the combination of easy-to-use software packages, increased computer capabilities at reduced cost and reasonably priced digital color graphics terminals, comprehensive graphics capabilities are becoming cost effective for a great number of users which only a short time ago could not afford them. And if the decade produces low-priced color hard-copy devices, an even greater surge in computer graphics usage can be expected.

Today's computer graphics software packages fall into three main categories: computer-aided design (CAD); geographic contouring and mapping; and business analysis and statistics (including histograms, pie charts and linear curves). Computer graphics makes it possible to continuously refine the design of a model and to redraw a chart or map using new or revised data without the need to redesign from scratch.

Engineering and manufacturing applications include the design of complex devices through three-dimensional perspective representations provided by finite-element model generators, electrical and hydraulic schematics, logic diagrams and floor plans. Geographic applications include seismic, demographic, geologic, thermal and pressure maps and diagrams. Charts or graphs can include smoothed or interpolated curves, log or semilog axes, line patterns and shaded or patterned areas.

Easy to Use

Ease of use without the need for special training characterizes the new breed of computer graphics packages. Users can communicate with the packages from the viewpoint of their own disciplines or areas of interest rather than abstractly.

For example, using simple English-like commands, a manager can design and modify graphic representations quickly and easily at a display terminal and then have them printed out if desired. In some cases, modules are available that can adapt basic graphics software to meet the needs of specific industrial applications.

In addition to the new generation of low-cost color graphics display terminals, outputs of computer graphics packages can be produced on ordinary CRTs, standard terminals enhanced for graphics, drum or flatbed pen plotters; the more rudimentary bar charts or point plots can be output on high-speed line printers. Most of the packages are written in Fortran for wide adaptability among computers.

Production Control

When it comes to realizing an effective return on computer investment, operations management is caught on the horns of a dilemma. Despite pressure to achieve a high level of computer utilization, upper management will not accept the high utilization that would result from giving exclusive attention to user demand; likewise, deemphasizing user needs in order to reach computer utilization goals is also unacceptable. Attaining 95% CPU utilization for a shift is hardly a good alibi for missing the payroll deadline.

The solution, of course, is automated job scheduling and production control, which is becoming increasingly practical for the data center through the acquisition of a software package.

Not only has comprehensive job scheduling been the exception rather than the rule in computer installations, but establishing and maintaining an effective job schedule is all too often considered an impossible task by operations management. There are hundreds, perhaps thousands, of possible job sequences that could be run with the available resources. Finding the right one is a mind-boggling task.

Although computers have been used for many years to automate factory production control, only recently have any appreciable inroads been made in the implementation of adequate production control in the data center, or even for the computer itself. The large, complex computer system, with its enormous computational and process-

ing power, being controlled by operators (and many times by programmers) shuffling card decks (and hoping all the cards are present when needed) has long been a great anomaly in the DP industry. Fortunately, this ludicrous scenario is disappearing as more and more data center managers become aware of the answers available through software packages.

New or improved computer production scheduling packages have been receiving a good deal of attention recently, and for very good reasons. A sampling of the features offered by these packages reveals such gems as:

- Coordinated schedules for all workstations in the data center.
- Dynamic scheduling of jobs based on user-specified time intervals.
- On-line job inquiry and tracking.
- Prompting of originator if a job cannot be scheduled.
- Automatic job mix optimization.
- Automatic control of job setup, eliminating JCL handling and issuance of job setup instructions.
- Detection of abends and incorrect completions, with action taken to correct errors and continue checking for completion of data collection before processing begins.
- Automatic rerun control, coordinated with pre- and postprocessing events.
- In-depth feedback, which can result in more realistic schedules.
- Data base for production control documentation, with statistics, history and descriptive information.
- Planning and forecasting, including simulation of new hardware configurations and prediction of the effect of new applications.

It is hard to imagine a DP installation of any appreciable complexity not using at least a simple job accounting package to provide basic utilization statistics as well as user chargeouts. Users of such high-concurrency systems as MVS could not operate with any reasonable degree of efficiency without comprehensive performance evaluation and feedback facilities.

Many installations, of course, fall between these two extremes. One of the most difficult tasks for managers of these intermediate-sized installations is deciding how deeply to become involved in performance evaluation and exactly what results to expect from the process.

What is needed — and what can readily be obtained through acquisition of software packages — is a comprehensive method of resource accounting that will give the DP department control over its own environment while providing a vehicle for budgeting, cost accounting and cost recovery. The package must measure and report system resources utilization and work load characters in detail so that performance problems can be solved and the system timed for increased thoughput and improved service to users.

Extending beyond the purview of performance evaluators are the sophisticated performance prediction systems, which can answer "what if" questions occurring in such activities as computer capacity planning, performance timing and system design. These packages can be used to support projection and balancing of work loads, determining when system capacity will be exhausted, calculating the effects of alternative upgrades and evaluating alternative vendor components.

DASD Space Management

One of the most common problems confronting DP managers is the seemingly insatiable demand for more disk storage. How can these demands be met in the most effective and economical way? Given the exigencies of random data storage, it is inevitable that some disk space will be wasted; but how much of a wastage factor should be tolerated?

Certainly, everything reasonably possible should be done to assure optimal utilization of existing space before requesting additional hardware. Efficient utilization, however, is not the whole problem — security, as well as backup and archiving, must also be effectively handled.

A large number of systems software products are currently being offered to assist in DASD space management. This is probably the most heterogeneous of all the systems software categories and, therefore, presents the greatest difficulties in evaluation and selection. The DASD space management problem can be approached in various ways — no two packages solve the problem in exactly the same manner.

In the following sampling of features and benefits, it is evident that a number of advantages can be realized in addition to basic space management. DASD space management packages:

- Increase disk space availability by as much as 50%.
- Eliminate space fragmentation.
- Reduce multiple extents.
- Highly compress data.
- Automatically reorganize volumes for optimal performance.
- Automatically reblock files.
- Improve multiprogramming performance.
- Eliminate operator errors and reruns due to exceeding disk storage limitations.
- Dynamically allocate scratch disk working space from a space pool.
- Automatically migrate and recall data sets.
- Support interactive programming systems (TSO, Roscoe).
- Display a catalog on-line, reflecting the status of all archive/backup files residing on off-line volumes.
- Display a pack map graphically.
- Provide customized reporting facilities.
- Provide billings for disk space.

We have looked at applications development packages that allow an all-inclusive approach to building and

How to count your chickens before they hatch.

Surprises can be expensive. Even good news can cost money if your company is not prepared for it.

With financial modeling you can avoid surprises and plan calmly for whatever the future has in store.

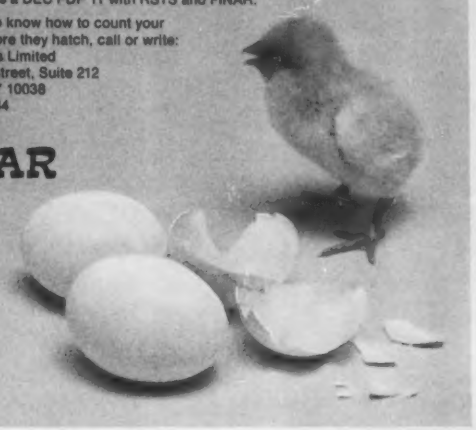
FINAR is the latest financial analysis and reporting system. It will help you plan:

- Budgets
- Cash flow
- Capital investment
- Project evaluation
- Forecasts
- Consolidation

All you need is a DEC PDP 11 with RSTS and FINAR.

If you'd like to know how to count your chickens before they hatch, call or write:
Finar Systems Limited
132 Nassau Street, Suite 212
New York, NY 10038
(212) 222-2784

FINAR



IBM DOS USERS
DOSSIER FOR YOUR 4300

"We will use DOSSIER as our major tool in converting to the new 4300....DOSSIER will continue to help us efficiently manage our libraries.

"After trial use, we licensed DOSSIER and analyzed our libraries in one day! We are delighted with DOSSIER—it's a very fine product."

Debby Schumann, Systems Programmer, Purity Supreme Supermarkets, Massachusetts.

Debby Schumann had been managing her program libraries manually, until she acquired DOSSIER. Not only does DOSSIER automate her library management, it will be an excellent tool to help her convert to the 4300.

DOSSIER goes beyond source library documentation to translate object code from program libraries into complete and current program and file information. Its 16 management reports address all aspects of controlling and auditing object libraries.

So whether you have a 4300 conversion on the way or are trying to maximize the effectiveness of your recently installed 4300, DOSSIER will help you. Try DOSSIER free, without obligation, for 30 days. Discover why more than 600 companies are already using DOSSIER to upgrade, maintain, audit and clean up DOS, DOS/VS and DOS/VSE libraries.

For more information, write to Computer Concepts, Inc.
6443 S.W. Beaverton Highway
Portland, OR 97221.
Or call 503/297-4741.

DOSSIER®
Documentation & Standards Tool
a product of
CCI

We have looked at applications development packages that allow an all-inclusive approach to building and maintaining applications systems by reducing reliance on the traditional programming process. These systems, however, will never eliminate the need for a certain amount of skilled programming, and the predictions for an almost insatiable demand for software in the '80s clearly indicate that "traditional" programming will continue to constitute a major time and cost factor.

The wide range of program development aids currently offered by the systems software vendors can do much to streamline the programming process. For the purpose of this discussion, we will divide program development aids into the following categories:

- Conversational programming facilities.
- Cobol writing aids.
- Alternative languages.
- Testing and debugging aids.
- Translators and converters.
- Documentation generators.
- Microcomputer programming aids.

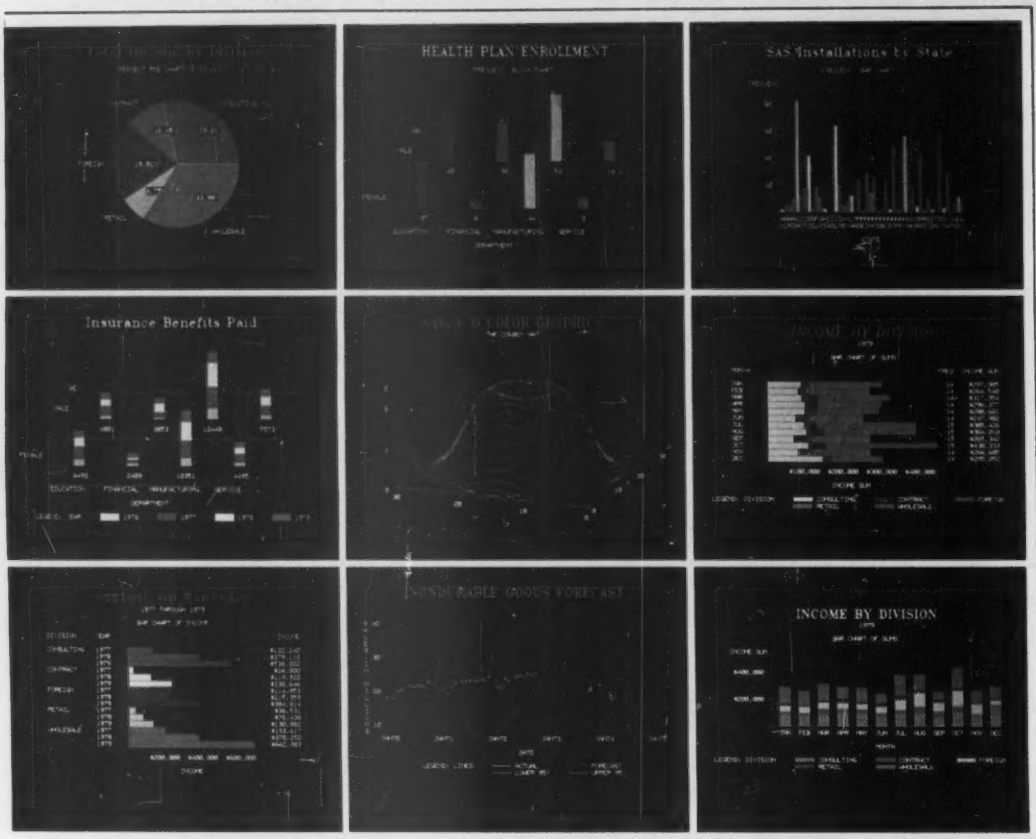
It is a well-established fact that programmer morale and productivity are increased through the use of conversational programming systems.

Particularly for those writing Cobol programs in the traditional way, Cobol coding is time-consuming, repetitive and stultifying. This condition contributes to the incidence of clerical errors, the correction of which further fuels the problem. Software packages are on the market that can reduce much of the verbiage associated with Cobol coding through the use of an abbreviated "shorthand" language and a special precompiler, producing more efficient and standardized code in the process. In addition, such features are available as a macro processor that allows the generation of entire program segments for the development of new programs, test data generation, run-time debugging, generation of performance tuning information and printing of formatted reports that replace dumps.

Using alternative languages is another solution to the programming time problem; however, this can be disadvantageous, since it often involves breaking with traditional standards and entrenched positions, and making a commitment in an entirely new direction.

Nevertheless, it may be a very worthwhile investment. Replacement languages for such high-level languages as Cobol require less expertise and only a fraction of the programming time of the original language. In addition, they compile much faster and provide better file-updating, table-handling and report-generating capabilities. DBMS interfaces are an additional benefit.

Even a high-level replacement for assembler that produces comparably ef-



SAS saves time-graphically.

Designing software to save time is a SAS tradition. And we've packed our years of experience into a new product. SAS/GRAPH.

SAS/GRAPH pioneers a new direction for graphics software.

Old software systems came into being before hardware had much in the way of time-saving features. And because the software had to do most of the work it was designed to be device-independent.

But SAS/GRAPH is device-intelligent. It's designed to use the time-saving features built into most modern hardware.

For example, look at the blue slice of the pie chart above. SAS/GRAPH used one "pie" command to color it. A device-independent system would require hundreds of "move and draw" commands. With SAS/GRAPH you save computer time and drawing time.

And you'll save user time. Because SAS/GRAPH produces most displays with two or three simple statements.

But best of all SAS/GRAPH is integrated into SAS, so you'll get the benefits of the total SAS system – a system now saving time in report writing, statistical analysis, data management and as a higher-level programming language at more than 1,200 OS, OS/VS and VM/CMS sites.

And you can add SAS (including SAS/GRAPH) to your dp staff for less than half the cost of a new programmer.

Call or write today. We'll provide more information on a very graphic time-saver.

SAS Institute Inc., PO Box 8000, Cary, NC 27511. 919/467-8000.

Images produced by SAS/GRAPH with Tektronix 4027, Dunn Instruments 631 Color Camera System and Polaroid film.

ficient object codes, can call programs written in other languages and is highly portable from one computer system to another is available.

Testing and debugging programs is one of the more onerous tasks with which programmers must contend. Ensuring that all aspects of a program are fully checked out is a very difficult and time-consuming job, yet an installation pays a high price for going into production with poorly tested programs. With the increasing popularity of DBMS, testing requirements have become more critical than ever.

One of the major problems in this process is the difficulty of developing test data that will exercise every aspect of program operation. Fortunately, there are reasonably priced software packages that generate test data and rigorously test programs, thus permitting programmers to concentrate on more productive tasks and drastically reducing the incidence of program bugs in production versions.

Typical functions that can be performed by program-testing software packages include:

- The generation of flexible test files to test programs written in any language.
- The output of reports on execution characteristics of Cobol programs.
- The generation of reports on execution activity determined by the user's symbolic procedure names.
- The checking of output files and the documentation of errors.
- The production of all types of files and record structures.
- The automatic changing of field values.
- The extraction and incorporation of data from live files.
- Interfacing with DBMS.
- The embedding of test parameters in Cobol programs.
- The use of English syntax, Cobol-like parameter statements.
- The generation of multiple formats in any order and their repetition a variable number of times.
- The testing of one or more modules any number of times during a computer run.
- Interactive debugging facilities.

For those still operating with programs written in outdated languages or with languages that are associated with a computer no longer in use and consequently are being run inefficiently through emulation techniques, it would probably be worthwhile to either acquire a translation package or have the translations performed for a fee by a software vendor. Packages and services are available for converting various assembly languages to Cobol or PL/I, PL/I to Cobol, RPG to Cobol or PL/I or from one Cobol dialect to another. The DOS-to-OS packages, which can do much to alleviate the trauma of making this conversion, also deserve consideration.

Systems and program documentation continue to be the Achilles' heel of DP. Documentation is considered an onerous, time-consuming task. Further, since it is manually prepared and maintained, documentation can rarely be relied upon as complete and up-to-date. It usually does not fulfill its purpose even if it is complete.

The only effective solution to this dilemma is to generate as much of the documentation as possible automatically as a by-product of the development and implementation process. Software packages designed to assist in the production and maintenance of documentation range from program flowcharters to generators of complete applications systems documentation.

Retrieval, Analysis and Reporting

Perhaps the most widely used systems software package is the data retrieval, analysis and reporting system. Until recently, these packages were known as report generators but, in their present state of development, this simple term no longer adequately describes them.

The basic purpose of a data retrieval, analysis and reporting package, as well as the justification for its acquisition, is the production of formatted outputs (reports or files, or both) on a quick turnaround basis, without requiring a formal programming effort. Requests for reports or for file maintenance operations can be made directly by the end user through fill-in-the-blanks forms provided with the package or through the use of nonprocedural, English-like, free-form statements. Input is usually either parameterized or keyword-oriented.

Since purchase prices range from less than $2,000 up to $40,000 and more, it is apparent that a wide range of capabilities can be expected from the various data retrieval, analysis and reporting packages on the market. Features and capabilities that extend the basic functions and can be found in one or more of these packages include:

- Multiple requests per pass.
- File updating capabilities for straightforward maintenance procedures.
- Processing and interrelating of multiple files.
- Automatic report formatting (with override).
- Extensive statistical capabilities.
- Cross-tabulation.
- Graphics output.
- On-line I/O and interactive options.
- Special features for the auditing function.
- Interface with one or more DBMS.

There are two fundamental types of data retrieval, analysis and reporting packages: the load-and-go, or interpretive type and those that require a preprocessing step. It is important to be aware of this distinction when considering acquisition of this kind of package.

In general, the interpretive type is preferred where ad hoc reporting requirements predominate. As might be expected, however, many trade-offs

Report writer software doesn't have to be expensive to be the best buy.

Compare Costs*	[redacted]	$450.00 Month	$560.00 Month
Compare Features	DYL-260	"B" Product	"C" Product
Full Function Utility	YES	NO	NO
Multiple Reports	YES	YES	YES
Automatic Report Composition	YES	YES	YES
Supports All Operating Systems At No Additional Cost	YES	NO	NO
Data Base Interface Option	YES	YES	YES
Disk Support Including FBA	YES	YES	YES
File Definition Not Required	YES	NO	NO
Over 1,000 Installations	YES	YES	YES
Member 1979 Datapro Software Honor Roll	YES	YES	NO

You can afford to buy the best. So, why don't you? And save money at the same time. Call or send for information.

Dylakor **(213) 995-0151**

16255 Ventura Blvd.
Encino, CA 91436

For 360/370/4300 and compatible computers. Additional monthly rental and lease plans are also available.

*Monthly costs on three-year contracts. Data from survey completed 6/15/80; subject to change.

IBM can't sell you the best programming language for the 4300 series. At any price.

HENCO CAN.

It's INFO—at far less than you'd pay for less capability—the plain english, high level language that's now in use in over 250 installations. Managers, Programmers and Secretaries agree, it's simply the best software available for IBM 4331, 4341, 303X and 370 with VM/CMS; Prime,Honeywell Level 6 or DEC VAX. In plain english, INFO will do it all:

● Data Dictionary ● Data Entry ● Data Update ● Data Retrieval ● Query ● Report Generator ● Multiple Files ● Programming Language ● Relational Data Base

Write or call for the book that will open your eyes to what a great application development language can be.

INFO PRIMER

HENCO, Inc.
35 Walnut Street
Wellesley, MA 02181
(617) 237-4156

are involved. Users of several of the interpretive types of data retrieval, analysis and reporting packages feel that the products are efficient enough to be used for the majority of their production reporting.

The purpose of information storage and retrieval (IS&R) systems is to facilitate the retrieval of documents or text extracted from documents.

A typical sequence of tasks performed by an IS&R system as the information passes from originator to user follows:

• Originate — Information is generated and packaged into a document.

• Acquire — The information is analyzed and, when appropriate, a surrogate is created. The surrogate is a substitute for the document and normally contains a bibliographic citation (document number, title, author, journal, volume and date), abstract (brief summary of document contents) and extract (direct quotations from the document). The surrogate allows the user to survey the material quickly without actually retrieving the document.

• Index — A set of index terms that characterize the document is usually assigned and associated with the surrogate.

• Store — The material is stored for future use in a manner that facilitates retrieval. Creation of the index is completed by associating the location of the surrogate or document with each index term assigned to the document.

• Announce — Potential users are notified that the document is available at a specified location.

• Browse — The availability of the information can be determined by using the index or surrogates to perform preliminary interrogations.

• Search — A search is conducted based on a question posed by the user. The terms of the query are usually compared with the index terms to locate information. The query is often formulated using the results of the preceding browse.

• Select — In response to the search, specific information is physically correlated with the query and fetched from storage.

• Extract — The selected information is gathered and integrated into reports.

Functionally related to IS&R is the general class of software that can best be classified as "text management" products. Packages in this category are capable of performing functions ranging from basic word processing to comprehensive management facilities for textual data.

These products can be used as stand-alone systems when the sophisticated retrieval capabilities of an IS&R system are not needed. Alternately, they can be used as a text preparation facility for an IS&R system (at least one product is capable of direct interfacing).

Special features include graphics; a user-controlled "shorthand," or macro

STRADIS™

—the first system development methodology designed around the structured technologies

Many of the leading organizations which have used IST's seminars on Structured Analysis and Design over the past 3 years have urged us to develop a formal set of standards and procedures—a formal structured system development methodology—which would incorporate the structured techniques.

Chris Gane and Trish Sarson, IST principals and authors of "Structured Systems Analysis: tools and techniques" (pub. Prentice-Hall) have worked with several of our clients to consolidate many years of project experience into this package.

STRADIS (STRuctured Analysis, Design and Implementation of Information Systems), is the first generally available system development methodology to be *designed around* the *structured technologies* from the start. It integrates the intelligent use of data flow analysis, data dictionaries, logical data design, structured software design, structured programming, structured walkthroughs and top-down development.

To evaluate **STRADIS** attend a one-day seminar, to be held in:

Our seminar "STRADIS: A Structured System Development Methodology Overview" reviews the structured technologies, presents the STRADIS standards and procedures in detail, and examines typical project deliverables. You will be able to examine your own copy of the STRADIS master reference manual (which you would customize), and you will come away with a thorough understanding of what STRADIS will (and will not) do for your organization. A detailed outline of this seminar is available on request. Fee: $150

Washington D.C.	**Oct. 15**
Houston	**Oct. 17**
Toronto	**Oct. 23**
Minneapolis	**Nov. 3**
San Diego	**Nov. 8**
New York	**Nov. 10**
Atlanta	**Nov. 12**

To enroll, or for more information, call John McCann on (212) 586-1098 or return coupon

To: John McCann, Improved System Technologies Inc. 888 Seventh Ave, New York, N.Y. 10019

☐ Enroll____ person(s) on the STRADIS overview seminar in _____ City

Call me for their names on ( ) _____ Phone

Name _____ Organization

Street _____ City, State, Zip

"Despite the popularity of disk and mass storage devices, many computer installations still operate with large tape libraries."

capability, for such procedures as reduction of data entry operations, standardized formatting and repetitive functions; and a scheme that allows entry of user tags in a document (the tags are resolved and special, user-specified actions performed on them at document processing time). Output can be distributed to a variety of hard-copy devices, and photocomposition interfaces are available.

Library Management

Disk-based librarians, a widely used type of systems software, are primarily designed for the management of source programs, eliminating cumbersome card decks. These systems offer much-needed security for program maintenance and storage, substantially reduce the programmer time required for program maintenance and provide good audit trails for verification of program activity and status.

Source program librarians' typical benefits include:

- OS-like library management features for DOS users.
- Data compression for high disk utilization.
- Storage of test data and JCL sets.
- Security controls to prevent updating the wrong member and to prohibit unauthorized access.
- Updating based on specified conditions, with data string and substring substitutions.
- Comparison and reporting of differences between two modules.
- Processing of a group of modules based on common features.
- Storage and retrieval of a large number of prior versions of a module.
- Storage of an unlimited number of temporary updates for program development purposes.

Tape-Based Librarians

Despite the great increase in popularity of disk and mass storage devices during the past few years, many computer installations still operate with large tape libraries. Tape volume manipulation and maintenance will always be a critical problem for the managers of these installations. Not the least of the reasons for magnetic tape use is the need for file backup and archiving, and one of the most difficult problems facing operations management is the need to verify constantly the usability of historical and archival tapes. Fortunately, well-developed software packages are available that can provide complete control over even the largest tape libraries.

In addition to the basic functions of tape library management, these packages can provide such benefits as:

- Control of movement and storage of tapes in remote locations.
- Automatic return of tape volumes to a scratch pool.
- Prevention of wrong tape mounts and inadvertent overwriting.
- Sharing of tape drives across DOS partitions.

Given the tremendous number of functions that computer operating systems must perform and the essentially noncompetitive circumstances under which those systems are developed, it is not surprising that the resultant inefficiencies have spawned a variety of extension and enhancement packages. The potential for innovative development in this area is almost limitless and the benefits to be gained by users substantial.

With only a few exceptions, operating systems are supplied by the hardware manufacturer and, as such, are not usually considered software "packages." A notable exception within the mainframe area is the availability of replacement operating systems for IBM's DOS. These replacement packages have allowed the numerous DOS systems no longer supported by IBM to remain viable, and they have enabled DOS users to benefit from the many desirable features otherwise available only by making the conversion to OS.

In the minicomputer field, several replacement operating systems are available, the most prominent being Unix, which has virtually established a de facto standard for minicomputer time-sharing operating systems.

The wide variety of operating system

THE ULTIMATE REFINEMENT.
PL/I FOR YOUR MICRO.

IBM's told us about PL/I for years. Now, we're telling them something.

PL/I Subset G. Newly designed expressly for minicomputers by an ANSI committee of PL/I experts. All the best application programming features, refined to match contemporary programming practices.

PL/I-80™ for Microcomputers. Based on Subset G, PL/I-80 runs under our CP/M® and MP/M™ operating systems.

PL/I-80 Means Performance. In an independent benchmark, PL/I-80 was tested against 34 other high-level languages for 8-bit processors, and came out on top.*

PL/I-80 versus the best of several languages:

PL/I-80	1.00
FORTRAN	1.21
BASIC COMPILER	1.71
PL/M	1.71
PASCAL COMPILER	1.89
PASCAL INTERPRETER	17.07
BASIC INTERPRETER	34.57

*Execution time ratio, Eratosthenes Sieve

Subset G is a brand new language with a new following. DEC®, Data General, and Prime now support it, and the list is growing. There's no better way to protect your software investment.

PL/I-80 from Digital Research. Fast. Economical. Refined. With the programming power of minicomputer PL/I. The package includes compiler, run-time library, linkage editor and relocating macro-assembler, plus three comprehensive manuals. All for only $500. (Domestic USA price only)

Interested in Refinement?
Call or write Digital Research. Experts in refinement.

DIGITAL RESEARCH®
P.O. Box 579
801 Lighthouse Avenue
Pacific Grove, CA 93950
408-649-3896
TWX 910 360 5001

READY FOR ELECTRONIC MAIL?

Your company could be moving messages electronically today. Learn how to convert your existing EDP hardware and network to an in-house electronic mail system.

One-of-a-kind seminar is designed for both managerial and technical decision-makers. Full day of instruction includes state-of-the-art demo, guest lectures and take-home materials. Meet the experts in computer-based message systems and put their experience to work for you.

Register now: morning session $85, afternoon session $125, full day $195.

Seminar conducted by Bolt Beranek and Newman Information Management Corp.

BBN Information Management

SEMINAR

The Theory		The Practice	
9:00 AM	Automating Office Communications: The role of computer communications and electronic mail.	1:30 PM	Balancing Risks, Costs and Payoffs.
10:00 AM	Computer-based Message Systems: A look at today's technological alternatives.	2:00 PM	Getting Started in Electronic Mail: Avoiding pitfalls while managing success factors.
11:00 AM	The Future of Electronic Mail: Identifying trends, anticipating breakthroughs.	2:30 PM	ARPANET: Profiling 10,000 users of electronic mail.
11:40 AM	Questions and Answers.	3:30 PM	Guest Speaker.
		4:00 PM	Live Demonstration: Three generations of software.
		4:30 PM	Questions and Answers
Lunch		Cocktails	

Call TOLL FREE or write for complete registration information: CW Conference Management Group, 375 Cochituate Road, Framingham, MA 01701

800-225-4698

1980 LOCATIONS: BOSTON, OCT. 14. **NEW YORK,** Oct. 20. **WASHINGTON, D.C.,** Oct. 23. **SAN FRANCISCO,** Nov. 5. **LOS ANGELES,** Nov. 7. **CHICAGO,** Nov. 17. **DALLAS,** Nov. 19.

DATA BASE BEST SELLERS

at a 10% savings to CW Extra readers

***Q.E.D. Best Sellers:**

■ NEW! Data Base Techniques (600 pp.)	$55.00
■ NEW! Design Guide For Codasyl DBMS (140 pp.)	29.50
■ IMS Design & Implementation (234 pp.)	29.50
■ Data Base Management Systems (440 pp.)	48.00
■ Data Dictionary Systems (504 pp.)	85.00
■ Data Base Systems (320 pp.)	34.50
■ Data Base Monograph Series (10 reports, 1000 pp.)	100.00
Subtotal	
Less 10%	
Prepay your order and deduct another 10%	
Shipping and Handling	2.00
Order Total	

Q.E.D.

☐Bill me ☐Check enclosed Canadian orders add 15% or remit U.S. funds

☐Master Charge or ☐Visa (BankAmericard) Prepaid Price
Account No. ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ Expiration Date _____

☐Please send the complete Q.E.D. book catalog (I will still receive the 10% discount.)

☐Please send information on Q.E.D. Education (See Q.E.D. course announcement in this issue.)

Name/Title

Company

Address

City State Zip

Offer Expires November 17, 1980

Q.E.D. Information Sciences, Inc. • P.O. Box 181 • Q.E.D. Plaza
Wellesley, Massachusetts 02181 • (617) 237-5656

extensions and enhancements defies precise categorization; however, for the purpose of this overview we have classified them as indicated below:

Sort/Merge Utilities. Packages that replace the sort/merge utilities of the IBM OS and DOS operating systems have achieved great popularity and success. These packages provide dramatic reductions in elapsed runtime, in channel and device busy time and in the number of I/O accesses required. Very substantial reductions in disk work space requirements and savings in programmer time and effort are additional benefits.

For on-line systems, the best executing mode for a given set of circumstances is determined dynamically. Since sorting and merging data represents a substantial portion of the computer work load in most shops, these highly developed packages often pay for themselves quickly.

Output Spoolers. In such operating systems as IBM's DOS, output spooling is used to decrease problem program execution time by intercepting supervisor calls for output devices and writing the output into storage. After the program runs to completion, the data is retrieved from storage and printed.

This technique, however, is costly in terms of the auxiliary storage space required, the amount of time needed to print the data and the overhead incurred while output devices occasionally idle as the data is spooled.

Several well-developed software packages currently available were designed to make this process more efficient. These products generally use the same techniques as conventional spooling to transfer data to and from auxiliary storage, but they differ in the way in which I/O commands are handled and records stored. The newer packages either use their own supervisor or modify the operating system supervisor.

DOS Performance Boosters. The desire of most DOS users to resist the conversion to OS as long as possible has resulted in a number of products designed to improve DOS performance. Some of the techniques used by these software packages, which can boost DOS performance by as much as 30%, are:

- Continuous dynamic partition monitoring and sharing.
- Resident transients that eliminate transient loading.
- Supervisor deactivation of jobs allowed only if it will result in improved throughput.
- Dynamic measurement of CPU time used by each partition and automatic setting of task-selection priorities.
- The cataloging of copies of programs in the core image library and their execution in any partition.
- The retention of directories in core and the fetching of multiple library blocks in a single disk revolution.
- Notification to the operator or system scheduler of the jobs that should be run together for maximum computer utilization and an alert when the system is becoming overloaded.
- Multiple transient areas provided to improve superisor performance by eliminating single-transient-area bottlenecks.
- The reduction of SYRES contention by allowing users to place any self-relocating phases, B transients or CRT transients in real or virtual memory.
- OS-like load library facilities and job accounting and performance evaluations.

Replacement Access Methods. The inefficiencies of IBM's Isam and, to a lesser extent, Vsam have provided the motivation for developing several replacement access method packages. Processing time reductions of from 30% to 50% have been reported compared with sequential Isam operation, and even greater reductions have been found when operating in random mode.

Although Vsam is more efficient than Isam, it does have a disadvantage in that a large amount of prime storage is wasted when tracks are not fully utilized and the JCL required is quite complex because of the extensive cataloging needed. In addition, the core requirements are high.

Replacement access methods compare favorably with Vsam (especially with DOS/VS) since they do not require special cataloging, are portable from one system to another and provide greater channel efficiency, resulting in better throughput for jobs contending for I/O. Transparency allows savings to be realized with little, if any, adverse impact on either present programs or future programming efforts.

Access Control. The increasing commitment of sensitive data to computer files, with the associated need for fast, convenient access, is resulting in a data security problem for both new and old users. Although loss of valuable data is a cause for concern, unauthorized access creates even greater hazards.

Some operating systems offer rudimentary access control facilities; however, much more rigorous methods are needed, and these are increasingly being supplied by software packages that permit controlling and auditing accesses to data.

One of the best approaches eliminates the usual reliance on passwords, using instead a technique that operates on the interaction between an access identification string, generated by an algorithm selected by a security officer, and a protect code that is roughly analogous to a user ID, resulting in a much tighter security scheme.

Where data sensitivity is truly critical, the best solution is a package that provides encryption of the stored data and its keys.

By Burt H. Liebowitz

THE DIMENSIONS OF DISTRIBUTED PROCESSING

Level Of Cooperation	Geographical Separation: Close	Distant
None	(1) Centralized Independent Processors	(2) Decentralized System
Modest	(3) Local Area Network	(4) Distributed Network
Close	(5) Locally Distributed Multiple Processor System	(6) Geographically Distributed Multiple Processor System
Intimate	(7) Multiprocessor	Not Applicable

FIGURE 1. TYPES OF DISTRIBUTED SYSTEMS

	Minis	370 Upgrade
Hardware Costs (Amortized on a Yearly Basis)		
Computer	$120K	$100K
Terminals	20K	75K
Conversion	20K	20K
Communication Cost	5K	50K
Amortized Software	87.5K	62.5K
Total Cost Per Year	$252.5K/yr	$307.5K/yr
Total 4 Year Cost	$1,010K	$1,230K

FIGURE 2. COST COMPARISON

	Distributed	Mainframe
Hardware Lease & Main	$915,108	$1,353,999
Software License Fees	57,100	
Software Conversion (in-house)	70,065	
Software Conversion (in-house)	209,682	
Training	12,650	
Operations	95,326	
Power	90,326	
	$1,450,457	$1,353,999

FIGURE 3. COMPARATIVE COSTS FOR A DISTRIBUTED VS. MAINFRAME APPROACH

Computing needs have been primarily served until recently by centralized computing organizations divorced from the main flow of operations in an organization. Even when a user required relatively modest computing resources they were provided by the shared use of a large computer.

The dissemination of computers to the point of use was discouraged because of cost. Numerous studies indicated that significant economies of scale would result from the use of a single or relatively few large computers known as "mainframes."

Even where a single application was large enough to demand a dedicated computing facility, large mainframes dominated the computing scene. Here the need for raw computing power rather than pure economics was the dominant factor.

Now, the driving force behind the use of large mainframes and centralized computing facilites has somewhat eroded. Minicomputer technology has developed to the point where it is feasible to consider locating small computers at user sites. It is also feasible to tie together several closely cooperating minicomputers or microcomputers to perform dedicated applications. These approaches are potentially economical because of the low unit cost of minicomputers and microcomputers.

Popular usage has attached the name "distributed processing" to any approach that employs more than one processor to solve a computing problem. Hence we now have a conflict between advocates of distributed (or decentralized) computing and centralized (mainframe) computing.

The conflict has been going on for years, but the recent explosion of minicomputers has brought the debate out in the open.

In this article we will explore the dimensions of distributed processing, especially its pros and cons vis-a-vis the use of mainframes. We will start with a definition of distributed processing which encompasses a broad range of system architectures. We will then identify seven distinct concepts that follow from the definition. These concepts vary widely, hence we recognize immediately that distributed processing is not a monolith. Therefore, pros and cons cannot be discussed without identifying the application context.

The issues and evaluation criteria are so diverse that we can jump immediately to the bottom-line conclusion of this paper — to wit, distributed processing (in whatever form) is just another tool available to the system architect. Its applicability depends on the project, the organization to be served, politics, support needs, corporate data processing chargeback schemes and corporate courage.

We will define distributed processing as follows:

> *A distributed system is one in which the computing functions are dispersed among several physical computing elements. These elements may be colocated or geographically separated.*[1]

We distinguish distributed systems from conventional mainframe approaches wherein all computing functions are performed with one large computer.

The spectrum of distributed processing applications can be depicted as the two-dimensional matrix shown in Figure 1. One axis represents the geographical separation of processors. The second axis represents the degreee of cooperation between the processors.

Although both dimensions are continuous, for convenience we can group each dimension in finite categories. On the axis representing geographical separation, we recognize two categories: distant and close.

By "close" we mean processors that are in the same room, building or at most in buildings that are located within thousands of meters. This closeness allows us to use privately owned low-cost, high-bandwidth interprocessor links (1M bit/sec up to 50M bit/sec).

For greater distances we must use common carrier services. These allow data to be sent almost anywhere, but usually at bandwidths much lower than those available for closely located systems. Common carrier services are provided by twisted-pair wire, microwave and satellite. Propogation delay, especially for satellite links, can be significant relative to the length of the message. Elaborate networks may have to be developed to ensure reliable message delivery to large numbers of inter-connected computers.

Four Levels of Cooperation

The other axis represents cooperation between processors. By "cooperation" we mean the level of interaction between the processors. We distinguish four such levels in our model.

Burt H. Liebowitz is executive vice-president of International Computing Co., Bethesda, Md.

Portions of the material in this article will appear in his forthcoming book, Locally Distributed Multiple Processor Systems, *by Burt H. Liebowitz, to be published by Prentice-Hall, Inc., Englewood Cliffs, N.J. This material is published with the permission of Prentice-Hall, Inc.*

HP 3000
HARDWARE AND SOFTWARE

- SUCCESS-PROVEN PACKAGES installed and supported by their developers, Hewlett-Packard and Data Systems for Industry.
- FIXED PRICE AND SCHEDULE with hardware-software discounts and leasing available to qualified customers
- INTERNATIONAL SUPPORT FROM OFFICES IN —

Los Angeles | New Jersey | Dallas | Europe

ACCOUNTING
- Payroll, Personnel
- Labor Distribution
- Accounts Receivable
- Accounts Payable
- Fixed Assets
- General Ledger

MANUFACTURING
- Order Entry
- Bill of Material Processing*
- Inventory & Order Status*
- Master Scheduling & MRP*
- Shop Floor & Capacity Control
- Job Costing

*(MM/3000)

DISTRIBUTION
- Order Entry
- Multi-Inventory
- Shipping
- Billing
- Replenishment Planning
- Distributed Processing

DISCOUNTS AND LEASING
- Up to 24% volume discounts on hardware-software systems sales
- Leasing available for hardware-software system sales

DATA SYSTEMS FOR INDUSTRY
3450 E. Spring Street, Long Beach, CA 90806
(213) 595-8741

SERIES/1

RMS
RESOURCE MANAGEMENT SYSTEM
IBM'S IAM VS. RMS

	IAM	RMS
multiple indexing	NO	YES
dynamic creation of index fields	NO	YES
duplicate keys allowed	NO	YES
flexible disk space management	NO	YES
automatic recovery	NO	YES
built-in hash function	NO	YES
variable length records	NO	YES
special sysgen required	YES	NO
fully integrated data management	NO	YES
compatible with RPS	YES	YES

RTP
REITER TRANSACTION PROCESSOR

RTP greatly simplifes the preparation of interactive applications such as inquiries, data collection, order entry, etc. on the Series/1. RTP provides powerful online utilities for screen formatting, for data validation and for defining the flow of control based on function keys. RTP obviates the need for most application code commonly encountered for interactive applications. The entire application can be put together and demonstrated to the ultimate user in a matter of days (sometimes even hours). When necessary, short program segments (COBOL, PL/I, FORTRAN) can be added one at a time. All data management is handled by RMS.

RTP INCREASES PRODUCTIVITY OF PROGRAMMING STAFF !!!
RTP GETS YOUR APPLICATIONS UP AND RUNNING - FAST !!

Data Structures, Inc.
122 East 42nd Street
New York, N.Y. 10017

Call Colm Kelly
or Alex Hill
at (212) 697-9305

At the first level, we have no cooperation at all. Each processor is an independent computer. On occasion a magnetic tape might be used to transport data from one computer to another, but no direct level of communication exists.

To some, this category should not be considered a distributed system. However, to many organizations the issue of whether to off-load stand-alone applications from mainframes to minicomputers is significant, so we include this category for completeness.

The next level in the spectrum is that of modest cooperation. Here the processors do communicate. However, each processor performs a self-contained function and communicates summary results to one or more of its neighbors.

Communications could be batch-oriented. The processors could be connected by very low-speed, possibly dial-up lines. The failure of any one processor would affect only its area of operations.

At the next level we find closely cooperative systems. Here each processor regularly communicates with other processors and is dependent upon their support.

High-bandwidth links may be needed. A failure in any processor could seriously affect the mission of the entire system.

The highest level of cooperation we call intimate. Here the processors share memory, I/O, programs, data and tasks. These systems are commonly called multiprocessors. Their level of cooperation is so high that they can exist only in a local configuration.

Seven Categories

From Figure 1 we see that the matrix leads to seven categories of distributed systems:

1. Centralized Independent Processors (CIP). The CIP is a collection of independent processors located in close proximity. One or more processors may be mainframes; the others could be minicomputers.

An organization would choose this approach if it determined that it was more cost-effective to off-load applications to minicomputers than to augment or upgrade its mainframe.

Multiple computers also allow backup in the case of failures. Management, operational and maintenance problems could arise because of the use of different computers and languages.

2. Decentralized Systems. Here computers are geographically distributed throughout an organization to provide computing power at the point of need. The organizational subunits are independent and hence the computers are not linked. One computer center may house a large mainframe, the others may use minicomputers. This approach would be an alternative to a shared centralized mainframe with terminals at the local sites.

The major advantage of this approach is that local computers might be more responsive to local needs than a shared, centralized system. The disadvantage is the potential for duplication of effort and proliferation of resources. These problems can be minimized by the establishment of data processing standards and the application of effective management control.

3. Local Area Network. The computers in the local area network are located in the same room or building. They operate independently, but communicate with each other on occasion; hence, the computers are interconnected by a communications network. The network mechanism could be a message switch, shared cable or a collection of serial lines.

In general, the computers and terminals are under the control of one organization, easing management problems. If the number of processors and terminals are large, the design of the local network can become a complex task — a task even more difficult if the computers are heterogeneous.

Strict communications standards are required to ensure that the computers can process data passed between themselves. A network control center may be necessary to monitor the health and performance of the system.

The local area network could provide significant advantages in reliability, growth modularity and cost-effectiveness relative to a mainframe.

4. Distributed Network. The computers are geographically dispersed. They operate independently and communicate infrequently. The computers could be operated by different organizations and could be quite heterogeneous in nature. A major motivation for this type of network is the desire to share resources.

In this situation, significant management issues arise to ensure that effective communications can take place between the independent organizations. Technical issues concerning communications, translation of data, network operations and operational procedures could also arise.

5. The Locally Distributed Multiple-Processor System (LDMPS). The LDMPS consists of physically independent CPU-memory pairs (called processors) interconnected by physical communication links. The processors are in the same room or building so that the links can be built with high bandwidth and high reliability. The processors can therefore cooperate closely to solve a single problem or closely related group of problems. The processors tend to be in the minicomputer or microcomputer class.

The LDMPS has attracted increasing interest because it has potential advantages of cost-effectiveness when applied to real-time computing systems, especially those that require high reliability.

This system architecture has been used in a wide variety of applications. In the commercial field these include

"The pros and cons of distributed processing vary between the seven classes of systems."

on-line banking, theater and sports reservations, process control, funds transfer, message switching and wagering systems. In government, they include spacecraft control, air traffic control, vessel traffic monitoring, weapons system control and missile tracking.

Technical issues usually dominate in the design of the LDMPS.

6. The Geographically Distributed Multiple-Processor System (GDMPS). This architecture is similar to the LDMPS except that processors are geographically separate, even though they cooperate to solve the same problem. These systems are limited in applicability, because of the narrower bandwidths available and longer delay times associated with interprocess communications.

This architecture would be applied when the user group is geographically dispersed. An on-line banking system, for example, could be established with processors and files in local communities. The data base would be restricted to local customers.

However, all local systems would be interconnected so that an inquiry could be directed to a foreign site if the local data base did not contain the required information.

The design issues associated with the GDMPS are similar to those of the LDMPS except that communications resources are more restricted and must be provided by common carriers. Problems of maintenance may also be difficult to resolve in the GDMPS.

7. The Multiprocessor. The multiprocessor consists of several processors, each linked to common memory and common I/O. No processor is dedicated to a particular task. Failures of a processor usually result in loss of throughput, but not functional capability.

The multiprocessor can be used for the same applications as the LDMPS and can therefore be considered an architectural alternative to it. Multiprocessors have been around for a long time and have been extensively discussed in the literature.

Significant issues of hardware construction, operating system design and system checkout exist. It is hard to build multiprocessors and very few off-the-shelf systems exist. Most multiprocessors have been built specially for specific projects or are extensions of large mainframes. Multiprocessors have limited expansion capability because of contention for shared memory.

Pros and Cons

The pros and cons of distributed processing vary between the seven classes of systems. To focus the discussion, we recognize two major divisions covering the seven classes of systems.

The first division encompasses those systems dedicated to the support of a demanding real-time application. We call these Category A systems. These tend to be served by architectural classes 5 through 7 in Figure 1.

The second division encompasses those systems involved with shared applications. The computer architecture is designed to serve multiple users who are relatively independent of each other. We call these Category B systems.

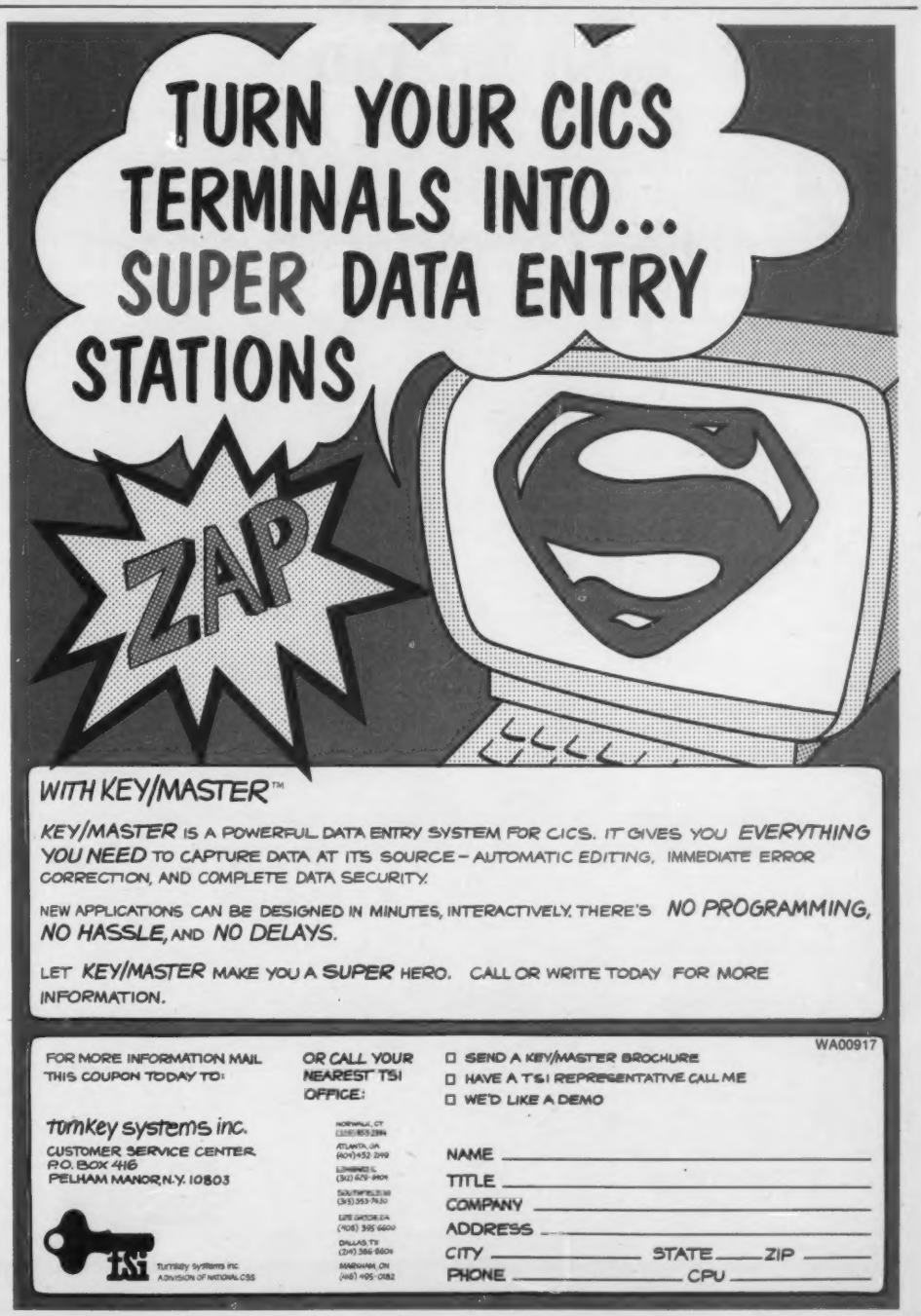
Category B systems tend to be served by architectural classes 1 through 4 as identified in Figure 1. We will consider each division separately.

Category A Systems

Category A applications are recognized to be technically demanding. They usually require fast response time to input. High reliability and high throughputs are not uncommon. So the decision has been made to dedicate a system to the application.

The system is usually under the control of one organization, most commonly the user group; thus, management issues associated with organizational structure are not a concern.

The architectural choices are to either use an adequately sized mainframe or a distributed approach encompassing either a multiple-processor system (MPS) or multiprocessor. Of the two distributed approaches, the MPS seems to be attracting great favor.


TURN YOUR CICS TERMINALS INTO... SUPER DATA ENTRY STATIONS
ZAP
WITH KEY/MASTER™
KEY/MASTER IS A POWERFUL DATA ENTRY SYSTEM FOR CICS. IT GIVES YOU EVERYTHING YOU NEED TO CAPTURE DATA AT ITS SOURCE – AUTOMATIC EDITING, IMMEDIATE ERROR CORRECTION, AND COMPLETE DATA SECURITY.
NEW APPLICATIONS CAN BE DESIGNED IN MINUTES, INTERACTIVELY. THERE'S NO PROGRAMMING, NO HASSLE, AND NO DELAYS.
LET KEY/MASTER MAKE YOU A SUPER HERO. CALL OR WRITE TODAY FOR MORE INFORMATION.
WA00917
FOR MORE INFORMATION MAIL THIS COUPON TODAY TO:
turnkey systems inc.
CUSTOMER SERVICE CENTER
P.O. BOX 416
PELHAM MANOR, N.Y. 10803
TSI
Turnkey Systems Inc.
A DIVISION OF NATIONAL CSS
OR CALL YOUR NEAREST TSI OFFICE:
NORWALK, CT
ATLANTA, GA
(404) 452-2919
(312) 679-1909
SOUTHFIELD, MI
(313) 353-7630
LOS GATOS, CA
(408) 395-6600
DALLAS, TX
(214) 386-8604
MARKHAM, ON
(416) 495-0182
SEND A KEY/MASTER BROCHURE
HAVE A TSI REPRESENTATIVE CALL ME
WE'D LIKE A DEMO
NAME
TITLE
COMPANY
ADDRESS
CITY
STATE
ZIP
PHONE
CPU

Hence we will compare the MPS against the mainframe.

Many applications of the type described as candidates for MPS solution have been implemented on mainframe architectures. By mainframe we mean large computers, exemplified by the IBM 370, Univac 1100, Control Data Corp. 6000 series and others.

In the past, a major deterrent to the MPS was cost. A collection of small computers equivalent in capability to a large computer was more expensive than the mainframe.

However, processor technology has changed. Low-cost computing power in the form of minicomputers and microcomputers presents an increasingly attractive alternative to mainframes for many computing tasks, and this trend will continue in the future.

To make this point we will discuss the advantages and disadvantages of the MPS relative to mainframes and provide some examples of the cost-effectiveness of the MPS in specific applications.

Advantages of the MPS

With falling costs, the potential advantages of the MPS can be applied to a wide array of processing problems. These advantages include reliability, response time and modularity:

- **Reliability.** All computing systems will fail. Depending upon the application, downtimes resulting from failures can be tolerated for varying amount of times. A hardware failure may take hours to fix.

Overall availability for mainframes has been estimated at 98%. In applications where higher levels of performance are required, redundant equipment must be provided. This usually entails the duplication of all equipment.

Redundancy

The MPS offers the advantage of low-cost hardware redundancy, since it may not be necessary to duplicate all the equipment to achieve the desired degree of availability. Consider a system in which 10 repairable processors are required to support system functions. It can be shown from the techniques described in the literature that the addition of just a few spares will provide extremely high availabilities and a large mean time between system failures.

Simpler Structures

In addition to hardware reliability, the MPS allows the use of simpler software structures than a mainframe with a complex multiprogramming operating system. Simpler software is more likely to be error-free than complex software and, hence, more reliable.

This simplicity follows from the dedicated nature of the MPS. Complex mechanisms are not required to protect unrelated users from each other. Because the functions are distributed between processors, the program size per processor tends to be smaller creating more manageable structures.

THEY SAID IT COULDN'T BE DONE.

we DID IT.

The big outfits couldn't do it. The little ones didn't even try. But we knew there had to be a way for minicomputers to communicate in a multi-drop environment without altering existing software. Or buying costly new hardware.

And we found the solution! The GoBetween Translation 200. A surprisingly inexpensive polling multiplexer and front end processor. Now your minicomputer can behave like the larger mainframe systems. The GoBetween Translation 200 allows up to 16 dumb terminals to operate synchronously or asynchronously at speeds up to 9600 bps sharing a single telephone line.

A POLLING INTERFACE FOR YOUR MINICOMPUTER.

It uses a simple Alpha addressing scheme, a Block Check Character for error detection and an "n" times retry for error recovery. Data is moved from the minicomputer to the remote terminal in blocks up to 256 characters and from the terminal to the mini in blocks up to 132 characters.

The GoBetween enables a minicomputer user to cut circuitry cost by transmitting more information over the same number or fewer lines.

Contact us today to learn more about upgrading your minicomputer communications capability. You'll be glad you did it.

Data Communication Systems by modemsplus

Modemsplus Inc. • 404 378-5276
217 E. Trinity Place • Decatur Georgia 30030

"MPS gives the user the ability to tailor system power to the response needs of the application."

• **Response Time.** The concept of response time is fundamental to the design of real-time systems. The definition of response time is "the time it takes from the initiation of an action until the time the computer produces an output."

In actual practice, response time is defined in a more precise fashion according to the actual application. In a theater ticket reservation system, for example, the definition may be "the time from the depression of the 'send' button on the terminal until a ticket starts to print."

The basic advantage that the MPS provides to the user is the ability to tailor system power to the response needs of the application. Another advantage relative to the mainframe approach derives from the single job dedication of the MPS; the user is not subject to the negative impact of unpredictable traffic loads caused by unrelated user groups.

Consider some of the ramifications of tailoring a system to user needs. A common problem in the design of real-time systems is that eventually — either because of underdesign or growth in traffic — the CPU approaches saturation. In a mainframe system this problem can be resolved only by a major redesign of software or an upgrade to a faster CPU, if one is available. In a properly constructed MPS, it will be relatively easy to add additional processors at a small incremental cost if more throughput is needed.

Another system bottleneck is disk accesses. An MPS offers the potential of distributing the data base across several processors, thereby reducing contention for disk accesses and channels.

• **Adaptability.** There exists a class of systems which perform similar or identical functions at a multitude of sites exhibiting a wide range of traffic flow. It is highly desirable to build such systems from a common architecture to minimize program development costs.

The use of a mainframe could be costly. Even if a single mainframe could support the load of the most demanding site, the same processor would be gross overkill at the smallest site. Costs are even more unbalanced if redundancy is required.

The MPS, on the other hand, could be tailored to site requirements in a cost-effective manner.

• **Physical Security.** It is desirable in some situations to distribute tasks among several processors that are separate from each other so that a physical failure in one room cannot affect processors in other rooms. Physical problems include fire damage, flood damage, sabatoge or, in the case of battlefield systems, enemy attack.

An example is a shipboard computing system used for weapons control and navigation. Instead of placing all critical computing equipment in one room, the major functions can be distributed to multiple computers located throughout the ship. The computers can be interlinked to back up each other so damage to any one computer would not affect ship operations.

An analogy in the commercial world would be to place processors in different rooms to protect against physical damage. For example, a message switch could be segmented into several small systems, each handling a part of the load. These systems can be dispersed into several rooms and interconnected by a communications link. The loss of any one room would at most affect only part of the load.

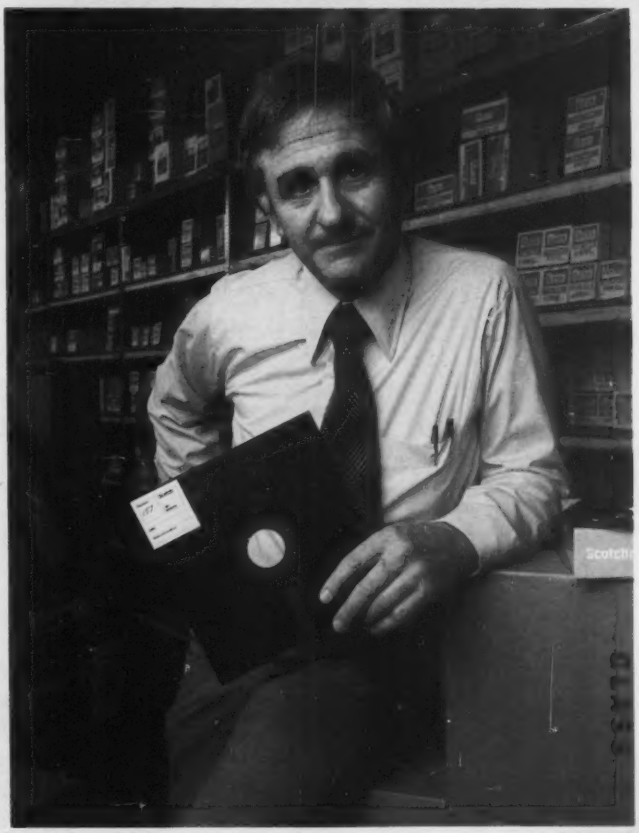
• **Program Modularity.** Certain types of MPS architecture enforce program modularity by dividing a system into functionally dedicated processors. Modularity improves program structure and programmer productivity and reduces checkout time.

Of course, modularity can be enforced in any system by means of effective management control. It is just easier where the physical design of the computing system enforces the concept.

Problems with MPS

The advantages of MPS over mainframes must be balanced by consideration of the problems associated with MPS, including the following:

"Our inventory is our existence. Think we'd trust it to anything less than Scotch® Brand Diskettes?"



Don Stone, President, Mass. Auto Supply Company, Inc., Boston, Mass.

Scotch Diskettes are the diskettes you can depend upon with the information *your* business depends upon.

Each one is tested and certified error-free before it leaves our factory. Because we know nothing less than perfection is acceptable for *your* vital business data.

Scotch Diskettes are available in regular or mini sizes, compatible with almost any system.

To find out where you can purchase Scotch Diskettes, call toll free: 800-328-1300. (In Minnesota, call collect: 612-736-9625.) Ask for the Data Recording Products Division. In Canada, write 3M Canada Inc., London, Ontario, N6A 4T1.

If it's worth remembering, it's worth Scotch Data Recording Products.

3M

• **System Inefficiencies** The limited address space of most minicomputers and microcomputers requires the use of memory mapping to manage large physical memories. Mapping can reduce the effective power of a processor by 5% to 20%, depending upon the processor and the application.

Increased operating system overhead caused by interprocessor communications can also reduce the effective power of the processor. This overhead is a function of the number of messages transmitted between processors per unit time and the physical transfer mechanism involved.

• **Complex Interprocessor Defects.** The MPS may be more difficult to check out and debug than a mainframe. With a single mainframe the system designer knows where to start looking when a failure occurs. In an MPS, the error could be in any one or more of several processors.

Worse yet, the symptom of failure may occur in a processor other than the one causing the failure. Couple this with the perverse nature of real-time systems — i.e., most errors show up in a transient manner and are related to unique and hard-to-reproduce traffic patterns — and we see that system checkout can be quite complex.

Although these problems can be minimized by extensive checking of interprocessor messages, it must be realized that error checking could add significant overhead to the system, thereby reducing overall performance.

• **Software Development.** More software development may be needed to build an MPS, compared with implementing the application on a mainframe. A major contributor to increased software development is the MPS operating system, which allows tasks residing within different processors to communicate with each other. Depending upon the hardware chosen, there may be little or no MPS operating system software available on an off-the-shelf basis. Even if some software is available, extensive error-checking routines may have to be written.

Also, software must be developed to allow individual processors to be checked in isolation prior to integration into a total system.

As a secondary factor, the designer may find less applications software available for small computers than for a large mainframe.

• **Lengthy Response Time**. The response time of MPS can exceed that of a mainframe even if the MPS is capable of supporting throughput requirements. This can occur for two major reasons: slower processors and inadequate interprocessor communications facilities.

• **People**. The development of a special-purpose MPS requires labor that may be in short supply. The systems aspect of the development may be quite extensive. Individuals who can understand hardware, minicomputers, communications and operating systems will be needed to successfully complete a project. It may take talent beyond that which is normally available in the organization responsible for system development.

This talent must be hired, trained or obtained by subcontract. In any case, this creates a problem of management and technical control.

• **Potential Maintenance Problems.** The maintenance of a geographically dispersed and possibly multivendor distributed system may be a significant problem. The potential for on-site maintenance is diminished as equipment is dispersed. A manufacturer may not have service centers near each site; on-call maintenance may be unreliable at best.

In some cases, these problems may be irreconcilable, so a centralized solution will be necessary. In other cases, a site failure may not be critical and slower maintenance response times can be accepted.

The multivendor problem can sometimes be resolved by employing the services of a third-party maintenance organization. In addition, the trend of the large-scale manufacturers to expand into the minicomputer field may alleviate some of the maintenance problems described above.

"Some startling breakthroughs have been made with respect to MPS cost effectiveness in the past few years."

A main consideration in the application of the MPS is its cost-effectiveness vis-a-vis the large centralized computer. Some startling claims and breakthroughs have been made with respect to MPS cost-effectiveness in the past few years.

Wirshing [2] considered a weather prediction system. He analyzed the cost-effectiveness of two computer architectures in the 1985 time frame. The first architecture was a machine designed along present concepts of pipelined arithmetic units, multiprocessors and arrays. The second architecture was an MPS network of microprocessors. Wirshing predicted a 40:1 price/performance advantage for the MPS.

Fuller [3] compared the C.mmp multiple-processor system with the Digital Equipment Corp PDP–10, a conventional uniprocessor. Fuller used two measures of comparison — millions of instructions per second (Mips) executed and processor memory bandwidth (millions of bits per second). Both measures were normalized on a per-dollar cost basis.

He concluded that the C.mmp outperforms the PDP-10 by a price/performance factor of 3.59 in instruction/sec/dollar and a factor of 2.78 in bit/-sec/dollar. These estimates were made on the average performance of four benchmark programs.

McGill and Steinhoff [4] claimed an even more imposing improvement in their analysis of an aerial combat simulation program. They compared the solution of the problem on an IBM 370/168 and on an MPS consisting of a minicomputer and 10 microprocessors. They claimed a factor of 500 improvement for the MPS in cost effectiveness.

Connell [5] described an MPS used to control a network of telescopes searching the sky for manmade objects. The processing load is high — on the order of nine Mips.

The proposed solution is a 10-minicomputer MPS connected on a common bus. Connell estimated that the system could be built at 10% of the cost of a large single computer.

The off-track betting system used in New York City is an MPS comprised of minicomputers. It replaced a dual IBM 360/50 configuration. The MPS provides four times the throughput of the 360/50 system and costs the City of New York $250,000 a month less to operate [6].

The Bank of America has built a system in which "the mini-computer hardware cost per transaction per second is about $50,000. This compares to about $300,000 in hardware cost per transaction per second for our former large-scale mainframe solution designed to meet similar uptime and reliability criteria."[7]

Of course, these claims must be carefully evaluated within the context of the problems being solved. Not all problems will favor the MPS approach.

Amenable MPS Solutions

It cannot be stated definitively which jobs favor an MPS or a mainframe. However, systems with the following characteristics seem to be amenable to the MPS approach:

- A need for guaranteed and predictable response time in a real-time environment.
- A need for high availability and reliability.
- High throughput.
- Extensive amounts of relatively simple calculations that do not require computers with elaborate instruction sets.
- An application with no extensive body of existing software on a mainframe.
- A problem with a natural division of subfunctions that are easy to assign to multiple processors.
- The ability of an organization to establish a knowledgeable management and technical team capable of supporting MPS technology.
- An application that can justify a dedicated computing system.
- An application that must be applied cost-effectively across a large number of sites, with a wide, dynamic range of throughput requirements.

Key to Success

Intel's integrated dictionary unlocks data management solutions.

Programmer productivity. Application development. Data base design. Project management. Programming standards. Maintenance. Data security, accuracy, integrity. Optimization of resources. Controlled growth.

All of these are priority issues among forward-thinking organizations today. But these are just a few of the issues addressed—and solved—by Intel's Information Resource Management product family.

The key is in Intel's Integrated Data Dictionary, IDD.

Get the big picture.

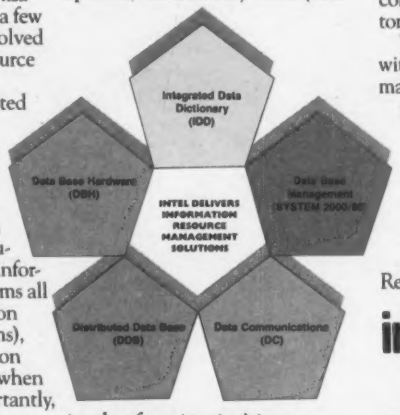
The IDD gives you a detailed, panoramic view of what's happening in your data base environment. By continually monitoring and tracking information (ranging from data items all the way up to entire application systems with multiple programs), Intel's dictionary puts reports on who, what, where, how, and when at your fingertips. Most importantly, you won't need a crystal ball to forecast the impact of change. Impact reports are standard, too.

Streamline application development.

With a wealth of vital information in hand, you can simplify the task of building and fine-tuning new applications. And because the dictionary works intimately with SYSTEM 2000®/80, Intel's powerful data base management system, designers are given extensive data modeling and application prototyping capabilities. Moreover, data items, records, and definitions are easily added, deleted, or modified even after a data base is in full production. And, like all IDD functions, this can be handled on-line or in batch mode.

Enforce standards, communicate, and cut redundancy.

Intel's data dictionary quickly tells you and anyone else involved in data base and application development, what already exists (and


Integrated Data Dictionary (IDD)
Data Base Hardware (DBH)
Data Base Management (SYSTEM 2000/80)
INTEL DELIVERS INFORMATION RESOURCE MANAGEMENT SOLUTIONS
Distributed Data Base (DDB)
Data Communications (DC)

in what form it exists) in your environment. Additionally, the dictionary applies editing, error detection, and correction functions.

Guarantee data security.

Watchdog of your data is IDD's sophisticated security system. With passwords which protect data as far down as the item level, you can be confident that your data remains "clean." You can be confident that unauthorized personnel will not accidentally alter or view sensitive information.

Implement successful solutions.

Intel's Integrated Data Dictionary is an application design aid, a documentation vehicle, a way to enforce standards and procedures, a master reference for determining the impact of change, a tool for controlling growth today and tomorrow.

To examine your key to success with Intel's IDD and other Information Resource Management products, clip the coupon below or call our Market Information Office at 512/258-5171. We'll respond with descriptive brochures, success stories about our customers, plus an invitation to our complimentary seminars on Information Resource Management.

intel® delivers solutions.

Europe: Intel Semiconductor (Nederland) B.V., Commercial Systems Division, Oranjestraat 1, 3441 Ax Woerden, The Netherlands, 31/3480-11264, Telex 47970 CSD NL

Canada: Intel Semiconductor of Canada, Ltd., 201 Consumers Rd., Suite 200, Willowdale, Ontario M2J 4G8, 416/494-8831

Please forward literature on Intel's Integrated Data Dictionary and other IRM products.

Name_____

Company _____

Mailing address _____

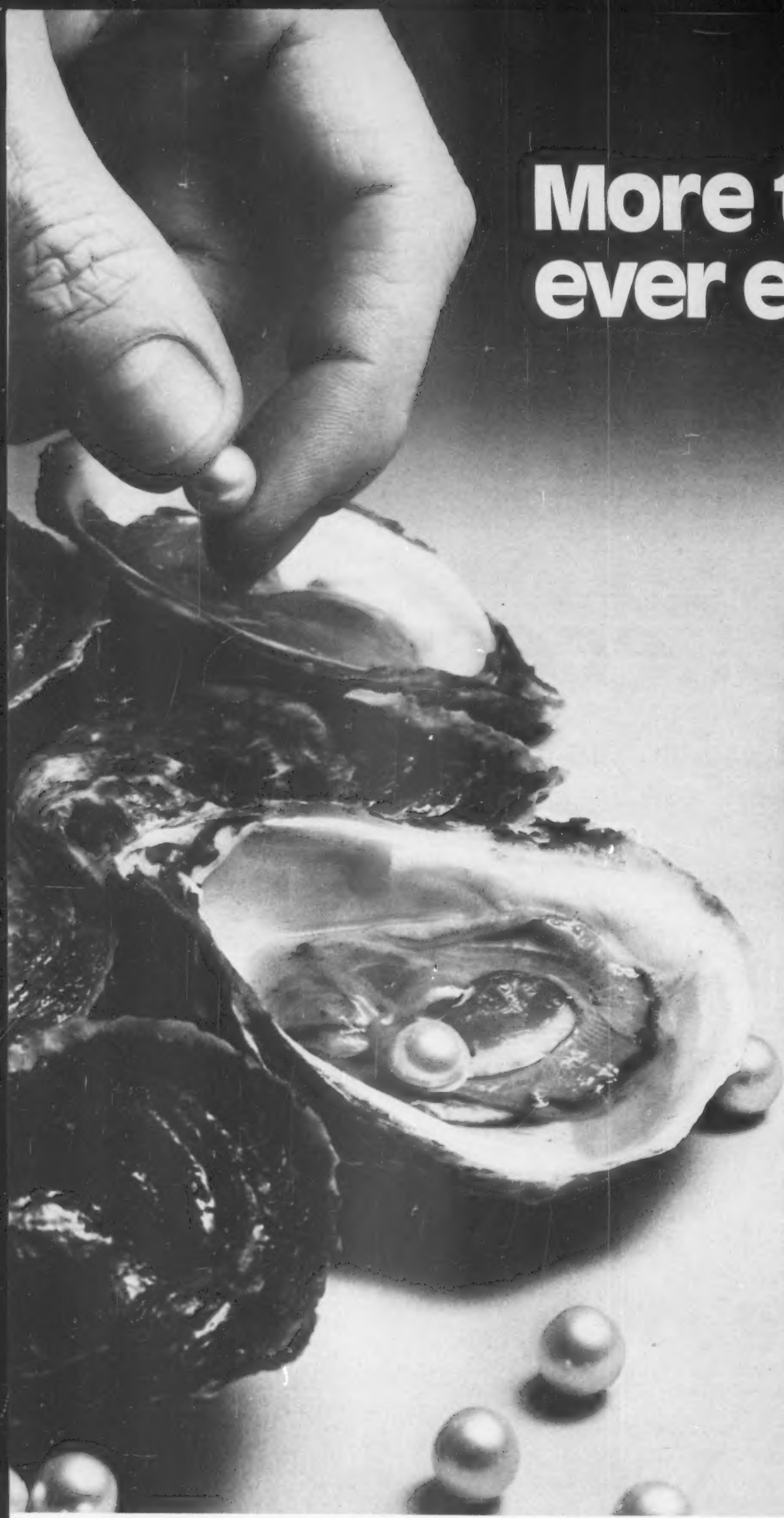
City, State, Zip _____

Telephone ( ) _____

Mainframe(s) installed: IBM (compatible) _____, CDC_____, Univac_____;

Operating system _____

Return to: Intel Commercial Systems Division
P.O. Box 9968; Austin, Texas 78766
ATTN: Market Information Office



More than you ever expected

Rare finds in software exclusively for BURROUGHS Medium Systems

SRI/EXPRESS — Two jewels in one setting. File maintenance module links existing data files to your CRTs . . . converts from batch to on-line without programming. Reporter module lets analyst or end-user generate reports without programming.

SRI/EDIT — Most widely used text editor in the Burroughs Medium Systems world. Member, *Datapro* 1979 Software Honor Roll. Release 5.0 adds time/event scheduling, security-by-user, and data file viewing to EDIT's well-known usability and ease of programming.

SRI/FIS-EGL — More than you ever expected in an on-line general ledger and accounting system. FIS-EGL adapts to your existing accounting conventions. FIS-EGL offers retrieval of any information in your system, on-line.

SRI
SYSTEMS RESEARCH INCORPORATED

2400 Science Parkway
Okemos, MI 48864
(517) 349-0200

Call (517) 349-0200 or send coupon

Ask Gail O. Wickham to show you how SRI software can make your Burroughs Medium System ever more useful, flexible. He'll send you detailed product information, arrange for demonstrations near you, or in our Data Center.

SYSTEMS RESEARCH INCORPORATED C9170
2400 Science Parkway, Okemos, MI 48864
Please rush further information on
_____ SRI/EXPRESS _____ SRI/EDIT _____ SRI/FIS-EGL

Name _____
Title _____ Phone _____
Company _____
Address _____
City _____ State _____ Zip _____
Burroughs System B2_____ B3_____ B4_____

"Until recently, the economic argument has favored centralized mainframes. With minis, the argument is no longer one-sided."

As system characteristics diverge from the above, a mainframe approach would be favored. This is specially true if the available work force is more comfortable with the mainframe.

Category B Systems

Category B systems support shared applications. These applications could be quite diverse. Some systems support time-sharing, batch and real-time transaction processing simultaneously. Management and organizational issues may be quite important since the user groups are usually organizationally distinct from the computer center. The organizations may have different goals, thereby creating organizational conflict.

The desire to have more control over their destinies has made distributed processing attractive to user organizations. On the other hand, economies of scale and centralized management control make centralized computing attractive to corporate staff.

Until recently, the economic argument has clearly favored the centralized mainframe approach. Now, with low-cost minicomputers, the argument is no longer one-sided. Under certain circumstances it may be more economical to split out applications and run them on stand-alone minicomputers under user control. We will look at the pros and cons of the distributed approach vs. the centralized computing center.

First, let us review the advantages and disadvantages of centralized processing. The strongest argument for centralized processing has been the economy of scale associated with the use of large-scale hardware and common software.

Centralization also provides improved efficiency in systems development programming. This derives from the homogeneous nature of the centralized computing facility. Put another way, programmers only have to be trained on one computer.

Centralization allows for better control of operations and standards because all computer resources are under the control of a single group. The use of common software makes it easier for users to share existing applications and support programs. These include data base management software, operating systems and retrieval programs.

Sharing reduces duplication of effort and enhances the value of scarce people resources. A large computer also provides capacity to process large complex applications. There may be some applications that are just beyond the capability of a minicomputer perform.

Training of new personnel is enhanced in a centralized facility. New people have to be taught fewer things. They interface with only one computer. This allows new people to be trained quickly.

In addition, it may be easier to acquire personnel trained in large computing systems compared with those trained in minicomputers.

All of these advantages must be weighed against the disadvantages of centralized computing. In general, a large centralized organization will provide less flexibility for tailoring programs to meet end users' needs. The organization would be most interested in optimizing on a global basis. This may cause inefficiencies for a particular user.

If users are geographically dispersed relative to the central facility, there may be high communications costs associated with the transmission of data.

A large computer can be expensive. It may be impractical to back up the system, so there could be a significant reliability issue associated with on-line applications.

Until recently this issue was not overwhelmingly significant. Most large facilities evolved from batch processing requirements. A user expected a long turnaround time; the additional time caused by the loss of the computing system would be hardly noticeable. In addition, batch systems are inherently recoverable because the input data always exists in the order in which it was processed.

With more and more users moving to on-line processing, however, the issue of reliability and recovery has become much more important.

Experience has shown that large computing facilities, especially those with large data base management systems, may have availabilities to a user in the 92% to 98% range even when redundancy is provided. A user which has moved from a batch environment might be willing to live with this performance for a while but, as users become dependent upon and familiar with on-line operation, the availabili-

EASTERN AIRLINES EDUCATION

PRESENTS

UNIVAC 1100 TRAINING

COURSES BEGINNING NOVEMBER 3

Some Courses Being Offered:

- ASET Programming
- ASET Internals
- CMS
- MASM
- DMS-1100 Programming
- USAS
- Programming Theory
- SSG

All courses are held in Miami, Florida

For more information regarding tuition and start dates, call collect:

GAIL FRASER
(305) 873-6526

"The disadvantages of minis are the potential for duplication of effort and loss of control in an organization."

ties cited will prove to be unsatisfactory.

Response time can be quite poor in a large computing facility. The computer system will be shared among many users. The load may fluctuate in unpredictable ways, and some users may be served slowly because of the impact of other applications.

Although mainframe computers tend to be more powerful than minicomputers, their operating systems have significant overhead. This overhead derives from the fact that the computer must protect users from each other and must also provide efficient access to expensive hardware. A single application may run quite slowly because of this overhead.

Other levels of overhead are associated with complex data base management systems which may be used even for relatively small applications. These systems may introduce many layers of disk access that are not inherently needed in the solution of the problem.

Another disadvantage of centralized processing is that a large incremental expansion is necessary once the computer becomes overloaded. The cost and impact of upgrading a mainframe can be quite significant.

We can now compare the advantages and disadvantages of decentralized processing. By decentralized processing, we mean the use of multiple small computers, each dedicated to one or a small subset of the total work load. We do not preclude in this definition the use of a large mainframe to support those applications that cannot be handled efficiently in a small computer.

Distributed computers allow data to be captured at the local level without the need for extensive telecommunications to a central computer. This data can be error checked locally prior to transmission to the computer.

Distributed computing allows applications to be tailored to the particular user. Since a lesser number of applications exist on any individual computer, the loss of an individual computer has less impact on overall performance. In addition, because minicomputers are inexpensive, it is possible to back up those minicomputers that do support critical functions.

Although the minicomputer is less powerful than the mainframe, minicomputer operating systems tend to be simpler. Minicomputers can therefore provide more computing time for the application.

Expansion of a minicomputer complex or a geographical set of minicomputers can be done at a smaller incremental cost than the expansion of the mainframe. Minicomputers can provide specialized and highly efficient file retrieval systems.

The disadvantages of minicomputers relate primarily to the potential for duplication of effort and loss of control in an organization. Unless management's standards are strictly enforced, there could be a tendency for each group owning and using a minicomputer to go its own way.

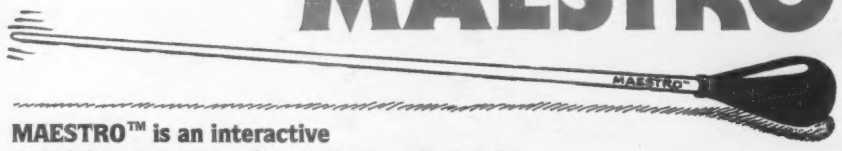
This could lead to a diverse set of equipment, increased maintenance costs and duplication of effort in what might indeed be common functions. This leads to more difficult control of system development, programming standards and data base.

In addition, minicomputers are limited in capability. It may be difficult to expand upward on any particular application. Some applications may run so long on minicomputers as to become effectively useless.

Operations might be more difficult in the minicomputer environment. Operators might be required at more sites and might have to be trained in more than one computer type. Facilities may have to be developed in more than one area. With each facility comes the problems of air-conditioning, power and security.

Most organizations that have successfully introduced distributed processing have done so with new applications that are relatively independent of existing applications and data bases. However, when applications are closely related to existing functions, it may be necessary to set up complex intercommunications networks and modify

Shackled by on-line programming systems? Take it off-line with MAESTRO™



MAESTRO™ is an interactive standalone programming system engineered to offload the mainframe and boost programmer productivity.

With Maestro you design, write, document and maintain your programs whenever you want, for as long as you want. You're free at last from the problems of CPU overload and slow response times. You enjoy the convenience of a dedicated Maestro terminal on a dedicated Maestro CPU devoted solely to **your** programming needs. The same goes for every other programmer in the shop—a separate terminal for each of you, with instant response times.

Maestro interacts with your target production mainframes through RJE. You can also use your Maestro terminal in 3270 mode for continued direct contact with the mainframes.

There's no other product like Maestro. And there's no combination of features like Maestro's anywhere else in the marketplace. Take a look...

Maestro Highlights

Interactive program design provides logic charts automatically for your systems and programs.

Multiple file support allows each programmer to have up to 12 files open and a position maintained in each.

Single-character keys generate often-repeated character strings.

Audit trails automatically track all program changes, so you can reconstruct any version at any time.

Word processing capabilities produce documentation in parallel with program development.

Maestro supports all phases of a program's life cycle including design, coding, testing, maintenance and documentation.

Memo file maintained for each user lets you send messages to each other via your terminals.

Print procedure gives hard copy printout, multiple copies and special forms if you wish.

Flexible procedure language enables you to adapt Maestro to any special development techniques.

Program development is offloaded from the mainframe; thus, you postpone or even eliminate the costly need to upgrade.

Programming proceeds independently of mainframe production crunches and downtime.

Maestro supports up to four communication lines. Each can have similar or different operating systems and/or CPUs, either RJE or 3270 passthrough.

CRT and keyboard for each programmer gives instant access to source code, project notes, documentation and other information.

USERS PRAISE MAESTRO

"Maestro is a timely and cost-effective way for us to remain competitive in the data processing marketplace and offer our employees state-of-the-art technology."

"These studies showed that by using Maestro instead of . . . , programmer cost effectiveness would improve by over 60 percent."

"There was absolutely no way our CPU would accommodate the number of terminals needed for our programmers. Maestro was the answer."

More than 3,000 MAESTRO terminals are now serving the DP community.

Call us today or use this handy coupon to request full details.

PLEASE RUSH ME A COPY OF THE MAESTRO BROCHURE.

Name _____

Title _____

Company _____

Phone ( ) _____

Street Address _____

City _____

State _____ Zip _____

MAESTRO™ SYSTEMS, INC.
Three Embarcadero Center, Suite 2470
San Francisco, California 94111
(415) 986-1188

Installing a software system by January 1?

We can help you.

Fiscal year 1981 is closing in pretty rapidly. That's why you need MSA's help now, to get the right management systems up fast and running strong for you in the new year.

We're part of MSA's 720 software professionals serving clients at over 4300 sites from Sydney to Seacaucus. We know that time is short and you're overworked; but that's why we're here. Just think of us as being on call to help you get the job done—properly and on time.

Once you've selected the MSA software system that's right for you—financial, cash management or human resources—we go to work. Our highly qualified installation and training experts work onsite with your staff to ensure a smooth conversion from your current system.

MSA's comprehensive learning aids are designed to provide a fast and thorough understanding that will help your people work with confidence, now and in the future. And our systems specialists are always available to answer your software and hardware questions.

MSA

MSA has a software system that's on target for your application needs. Send in the coupon below and we'll get you off to a running start in 1981.

MSA

3445 Peachtree Road, N.E.
Suite 1300
Atlanta GA 30326
attention: Robert Carpenter
Telephone: 404/262-2376

I am interested in the following MSA systems:

- ☐ Total Financial Management System—Big 8
- ☐ General Ledger/Budgeting—GL/80
- ☐ Payroll/Labor Distribution
- ☐ Accounts Receivable
- ☐ Financial Forecasting & Modeling
- ☐ Fixed Assets/Property Control
- ☐ Personnel Management & Reporting
- ☐ Accounts Payable
- ☐ Supplies Inventory Control & Purchasing
- ☐ Procurement Matching

Name	Title
Company	
Address	City
State	Zip
Telephone	Computer Model

CWEEK

existing programs.

If an organization is growing because of increased use of old applications, then decentralization may imply a significant amount of conversion. The cost and disruption of conversion may make it impractical to consider decentralization.

In summary, each case must be judged on its own merits. However, the technology is now in place and does allow distributed, decentralized computing to be considered by an organization. The ultimate decision should be driven by the needs of the organization.

Case Histories

We will discuss several case histories which illustrate some of the points made in this article. First we'll describe two Category A multiple processor systems; the Nasdaq securities quotation system and the Bank of America on-line inquiry system.

Each of these represents a successful implementation of a distributed architecture. The first illustrates the use of minicomputers to off-load a large mainframe-based multiprocessor. The second resulted from a trade-off analysis between large mainframes and multiple minicomputers.

We will also provide two Category B case studies. The first describes a distributed order entry and inventory control system. In the second, a local government computing center considered the use of centralized independent processors, but decided instead to upgrade its mainframe.

The Nasdaq System

The National Association of Securities Dealers (NASD) operates a nationwide telecommunications network that allows its members to rapidly determine the latest bid and ask quotations for over-the-counter securities. This system, called Nasdaq® — the NASD Automated Quotation System — was originally put into operation Feb. 8, 1971.

A major upgrade in 1979 enhanced the capability of Nasdaq's communications concentrators to support local data bases. This off-loads the mainframe and improves response time.

The upgrade takes advantage of the latest minicomputer technology and introduces distributed processing concepts to the network.

System Functions

Basic functions have remained constant since system inception. Each security in the over-the-counter market is assigned a group of market makers. It is the responsibility of the market makers to establish buy (bid) and sell (ask) prices for their security.

Before the introduction of the Nasdaq system, the process of buying and selling was manual. A trader wishing to buy or sell a particular security would have to telephone several market makers to determine the best price for his customer.

To shorten what could be a lengthy and confused procedure, many traders would call only a few market makers. The process was not conducive to providing the best price for the trading public. It was with the intent of rectifying his situation that the automated system was introduced.

The system provides traders with CRT/keyboard terminals. There are three grades of service provided by the system. A Level I terminal provides an inquiry capability for the current representative bid and ask price for any security in the data base.

A Level II terminal has Level I capability but is also capable of receiving the current bid and ask price for each market maker for a selected security. The broker can scan all prices for the best buy. The actual order is processed by direct telephone conversations with the selected market maker.

A Level III terminal is used only by market makers. It has all the capabilities of Level I and Level II terminals, but also allows the market maker to change bid and ask prices. The central data base is updated within seconds of the time a market maker enters a change.

From system inception until early 1980, terminals were connected to the

Computer Professionals...

Are you making as much money as you should be?

To find out, call today for our new local survey

It's FREE!

Includes 23 New Categories

The 1980 Survey has been expanded to provide specific data for 23 new categories. These include positions in data communications, mini/micro programming, data base administration, technical marketing support and EDP auditing in addition to our traditional categories of programming, software, systems design, management and computer marketing.

Do you know how your salary compares with other EDP professionals in your specialty—with the same level of experience—working in your city, or in other cities across North America? Are you being well paid for what you do? Are salaries better elsewhere?

Now you can find out simply by calling today and asking for a copy of our current 1980 Local Metropolitan Computer Salary Survey. It's FREE!

Source Edp, North America's largest recruiting firm specializing in the computer field, publishes this valuable Survey annually as a service to computer professionals from coast to coast. The 1980 Survey shows current salaries in your area, 37 other areas in the U.S. and Canada, plus overall national averages. Salaries are shown by career specialty broken down by length of experience or size of computer system.

As you will discover, this Survey can be extremely valuable in helping you make important career decisions. Call for your free copy today.

For your free report, simply phone the Source Edp office nearest you.

source edp

If unable to call, write
Source Edp, Dept. CE1
100 South Wacker Drive
Chicago, Illinois 60606

(When writing, please be sure to indicate home address and position title). Client companies assume our charges.

United States

Arizona	
Phoenix	602/258-9333
California	
San Francisco	415/434-2410
Palo Alto	415/856-0600
Los Angeles	213/386-5500
Van Nuys	213/781-4800
Torrance	213/540-7500
Irvine	714/833-1730
San Diego	714/231-1900
Colorado	
Denver	303/773-3700
Englewood	303/571-4450
Connecticut	
Hartford	203/522-8590
Stratford	203/375-7240
District of Columbia	202/466-5890
Florida	
Miami	305/624-3536
Georgia	
Atlanta Central	404/588-9350
Atlanta Suburban	404/325-8370
Illinois	
Chicago	312/782-0857
Northfield	312/446-8395
Oak Brook	312/986-0422
Rolling Meadows	312/392-0244
Indiana	
Indianapolis	317/631-2900
Louisiana	
Baton Rouge	504/924-7183
New Orleans	504/561-6000
Maryland	
Baltimore	301/727-4050
Massachusetts	
Boston	617/482-7613
Wellesley	617/237-3120
Michigan	
Detroit	313/259-7607
Southfield	313/352-6520
Minnesota	
Minneapolis	612/544-3600
St. Paul	612/291-2467
Missouri	
St. Louis County	314/862-3800
St. Louis Downtown	314/231-4880
Kansas City	816/474-3393
New Hampshire	
Nashua	603/880-4047
New Jersey	
Cherry Hill	609/482-2600
Edison	201/494-2800
Paramus	201/845-3900
New York	
Midtown	212/736-7445
Wall Street Area	212/962-8000
Long Island	516/364-0900
White Plains	914/683-9300
North Carolina	
Greensboro	919/378-1642
Ohio	
Cincinnati	513/769-5080
Cleveland	216/771-2070
Columbus	614/221-2070
Oregon	
Portland	503/223-6160
Pennsylvania	
Philadelphia	215/665-1717
King of Prussia	215/265-7250
Pittsburgh	412/261-6540
Texas	
Dallas	214/387-1600
Fort Worth	817/338-9300
Houston Central	713/751-0100
Houston Suburban	713/626-8705
San Antonio	512/344-0217
Virginia	
McLean	703/790-5610
Washington	
Seattle	206/454-6400
Wisconsin	
Milwaukee	414/277-0345
Canada	
Toronto	416/865-1125
Don Mills	416/425-5730

"The team recognized that functions could be off-loaded to the concentrators, thus improving mainframe efficiency."

system via concentrators located in San Francisco, Chicago, Atlanta and New York City. (While the data is from a 1972 reference[8], it is representative of the size of the system throughout its existence.)

The host site originally contained a dual-processor Univac 1108 mainframe working in a multiprocessor mode. The mainframe maintains the central data base for securities and market makers. All critical components are duplicated for reliability.

The mainframe performs the following basic functions:

- Updates the data base, using information received from market makers.
- Formats replies to Level I and Level II inquiries.
- Reports volumes and prices to the news media and other stock exchanges.
- Receives price and volume information from other exchanges for a selected set of non-NASD securities.

The system supports a data base of 6,000 securities. The response time for inquiry processing in the mainframe is approximately 8 msec to 50 msec. A query input is approximately 15 characters in length, a typical response is 115 characters. Peak transaction rates range from 70- to 100 trans/sec. These peaks occur at the time the markets open for trading and last for about one-half hour.

System performance for response time was specified as follows: 50% of all quote requests or quote changes must be serviced within 5 sec; 90% of these transactions must be serviced within 7 sec. The quote file must be updated within 5 sec, 95% of the time.

The concentrators are responsible for all telecommunications functions. They poll the OCUs, using a limited full-duplex protocol. All communications are error-encoded and checked.

The Enhanced System

The NASD has enhanced its system several times since inception. One major change was the upgrade of the original Univac 1108 mainframe to the Univac 1100/22. A Nasdaq study team concluded in 1978 that a major upgrade to the network concentrators would be desirable. This upgrade would encompass a replacement of the concentrator hardware and software and a change in concentrator site locations.

Several factors contributed to this decision. First, the concentrator hardware was becoming obsolete. Second, the original concentrator computers could not be expanded beyond 64K bytes of memory; precluding an increase in concentrator functionality. Last, traffic flow analysis showed that the Atlanta site was not cost-effective; hence, the Atlanta lines should be incorporated into the New York concentrator.

The study team further recognized that certain processing functions could be off-loaded to the concentrators, thereby improving mainframe efficiency. Concern for mainframe efficiency was driven by several factors.

Most immediate was the recognition that the system could bog down in peak periods if one of the two processors in the Univac 1100/22 multiprocessor failed. Although this was a relatively rare occurrence, it was of concern to the service-oriented management of Nasdaq.

The second factor was the recognition that the volume and type of processing in the network would be primarily driven by government regulations and pressure for a more fully automated market system.

The first phase of a plan to upgrade the system was announced in a request for proposals (RFP) released in November 1978. The RFP requested bids for a concentrator replacement. The new concentrators would be located in Chicago, San Francisco and Trumbull, Conn. The Chicago and San Francisco replacements would handle the same lines and functions and would each consist of two minicomputers.

The Trumbull concentrator would support the regional lines previously supported by the Atlanta and New York concentrators. It would be physically colocated with the Univac mainframe.

your wish is our KOMAND BY PACE

Do you wish you had:

- SMF data loss recovery
- CICS, IMS and VM accounting
- TSO and TONE billing
- Complete audit trail
- JCL error reporting
- Source code
- Guaranteed repeatability
- Total resource accounting
- Account number validation
- DASD billing and utilization
- Multiple charging rates by CPU
- Extensive debit & credit capability
- Discount & surcharge by shift, priority, and system degradation

Whatever utilization or billing features you would like in your system, KOMAND can meet most or all of your requirements. Not limited to just SMF, KOMAND puts you in control of the resources of your OS/VS system. KOMAND is cost conscious, flexible, and systems and operations oriented. 40% of our new clients had outgrown a competitor's job accounting package. When you start with KOMAND in the first place, it grows with your needs.

Send for a description of the KOMAND family of OS/VS job accounting products.

PACE Applied Technology, Inc. Telephone (703) 369-3200 TWX 710 663 0652
7900 Sudley Road, Manassas, VA 22110

☐ send brochures

Name _____ Title _____

Company _____ Phone _____

Address _____

City _____ State _____ Zip _____

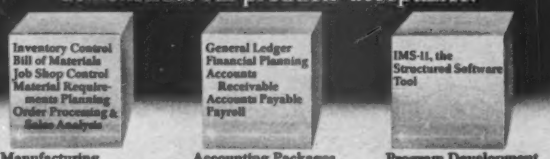
Computer _____ Operating System _____ We have: CICS ☐ IMS ☐ VM ☐

The PACE for Today . . . and Tomorrow

Award winning software packages for your DEC PDP-11

(operating under RSTS/E)

ICP Million Dollar Award and DEC AIP Gold Stars demonstrate our products' acceptance.



Manufacturing Packages

Tighter control over spiraling inventory costs, recognizing production backlogs *before* they develop and improved customer service are just some of the benefits of this set of packages. Some system highlights: multiple inventory costing methods, automatic rollup and recosting of bills, labor and work center utilization reports, shop load requirements and on-line "bucketless" MRP runs.

Accounting Packages

Our on-line, interactive packages provide you with immediate and *always* up-to-date accounting data. At a moment's notice you can verify customer credit, get account agings, pay vendors, and produce financial statements. Some advanced system features: multidivisional and multi-bank reporting, auto-cash application, variable dunning notices, A/P foreign gains/losses, budgeting, comparatives and G/L report writer.

Program Development

As a direct result of our staff's combined 100+ years of RSTS/E experience, the development of our applications under IMS-11 insures that all our software conforms to the same rigid development and documentation standards. As a stand-alone programming tool, development time of new systems can be cut up to 50%. Features include: programming standards, record I/O, keyed access, linked list and other data management functions, screen control, IMS sort utilities, menu and report directors, and JCL generation language.

Call (617) 489-3550 or write:

ims INTERACTIVE MANAGEMENT SYSTEMS
375 Concord Ave., Belmont, Mass. 02178

AUTHORIZED DIGITAL COMPUTER DISTRIBUTOR

DOS/VS (E) and CICS/VS

• • • CONSULTING • • •

- Systems Implementation Consulting
- Contract Systems Programming

Consultants: Ben Moyle, Minneapolis (see below)
Steve Huggins, 10317 Cherryview Ct., Oakton, VA 22124, (Wash. D.C.) 703-281-0222

Each consultant has 10+ years' experience in DOS/VS and VSE systems programming in multiple-CPU environments.

• • • PROGRAM PRODUCTS • • •

DOS/VS Terminal Printer Spooling

BIMSPOOL

Retrieves batch print from POWER/VS queue, converts and prints on 3270-type terminal printer via CICS*. May be used to eliminate RJE printers. $4000 or $200/mo.

DOS/VS(E) Library Display

BIMSERV

• NEW • Displays directories and entries in all DOS/VS(E) Libraries, and displays current contents of Label Area and disk VTOC's. An invaluable systems programmer tool. $1480 or $74/mo.

DOS/VS Console Hard Copy File Display.

BIMDSLOG

Used by operators or programmers to review DOS/VS messages via CICS* without tying up system console. $720 or $36/mo.

*Now available for SHADOW also.

DOS/VS(E) System Status and Queue Display

BIMMONTR

The original and still best DOS/VS(E) system activity, performance measurement, POWER/VS queue, and POWER/VS job output display system. (CICS) Numerous options to assist operators, programmers, and systems programmers. $920 or $46/mo.

Other products—BIMSUBMT—On-Line Job Submission—$1000 or $50/mo.
BIMDEVC—Device Assignment Status Display—$360 or $18/mo.

BIM B I MOYLE ASSOCIATES
4355 Lyndale Ave. S.• Minneapolis, MN 55409
612-822-2661

DOS/VS(E) and CICS/VS Consulting, Contract Systems Programming, and Program Products.

Member Independent Computer Consultants Association

In addition to the existing telecommunications functions, the Trumbull concentrator would contain a subset of the securities data base. Inquiries against that subset would be processed entirely by the concentrator, with no intervention by the mainframe. The mainframe would update the concentrator data base immediately upon update of its own data base.

Here's how it works. Inquiries are processed in the concentrator. If the inquiry is for the first frame of a locally stored security, the concentrator returns the response. If the inquiry is for a subsequent frame of locally stored data or for data that is stored only in the mainframe, the request is sent to the mainframe.

A response is formatted in the mainframe, sent back to the concentrator and subsequently transmitted back to the concentrator and subsequently transmitted back to the concentrator and then to the requesting terminal. Market maker transactions and non-inquiry transactions are sent to the mainframe. Data base updates are sent from the mainframe to the concentrator.

The Trumbull concentrator was specified to handle the following requirements:

1. All processing as described above.
2. 120 trans/sec peak traffic.
3. Up to 1,000 securities in the local data base.
4. No more than one hour cumulative downtime per year.
5. Sixty-four regional lines.
6. A total of 1,028 terminals.
7. Transaction processing time must be less than 10 msec.

A contract for the upgrade was awarded to International Computing Co. of Bethesda, Md., in January of 1979. Normally, input traffic is divided between two processors. A third processor acts as a roving spare and an off-line program development facility.

The processors are Univac V77-800 minicomputers configured with 768K bytes of main memory. (Univac V77-400s with 64K bytes of memory are used in Chicago and San Francisco.) This memory is large enough to store all programs, buffers and a data base for up to 1,000 securities.

Each processor has a keyboard/-printer console for entering commands and printing reports and error messages. Seven-track magnetic tapes are included for logging transaction records for those transactions processed locally. These tapes are processed by the Univac 1100/22 at the end of the day for the purpose of billing system users.

If one of these processors fails, its lines can be switched to the spare processor. If a second processor fails, all lines can be switched to the surviving processor. This may result in degraded response times. However, overall system availability is very high, since only one of three processors is required for system operation.

The concentrator is initialized by the mainframe at the start of the day. First the concentrator program is downline-loaded. Then a directory defining the local securities is transmitted from the mainframe to the concentrator.

When the markets open, a set of commands is sent from the mainframe indicating that trading can commence for each of the local securities. Data base updates can be sent from the mainframe at this point. The data base is therefore established in the concentrator over a period of time determined by the inquiry rate of the system. The initial update of the data base for a particular security will not take place until the first inquiry occurs for that security.

There are concentrator failure modes that could necessitate a recovery of the data base. One is the temporary loss of main memory; another is a hard failure that requires the switchover to the spare processor.

In either case, the recovery procedure is the same as that described above for system startup. The entire recovery procedure is controlled by the mainframe, thus simplifying the concentrator software and ensuring the consistency of the data base.

Bank of America

The Bank of America in California has installed a large multiple-minicomputer system to support on-line inquiries from teller terminals[7, 10, 11, 12]. The system was developed as an evolutionary adjunct to an existing and still functioning batch-oriented accounts posting system.

The need for the on-line system resulted from a recognition that real-time inquiry into savings and demand deposit accounts is essential to providing improved service to Bank of America's customer base.

A system study performed in 1973 determined that the existing batch system, based on IBM 370 processors, could not effectively be updated to include an on-line inquiry function. There was no manufacturer-supported software package that could operate both as an on-line system and as a batch posting system.

A follow-on study developed a system concept that would provide a front-end inquiry system to support the batch system. The system extracts an inquiry file from the previous night's batch file and loads the inquiry file into the front-end system. All real-time inquiry transactions operate against the front-end file.

System updates are captured in the front-end system and posted on a transaction log. At the end of the day, the updates are transferred by means of magnetic tape to the batch system, which posts the updates. The cycle repeats for the next day.

Early considerations for the architecture of the front-end inquiry system included the use of separate IBM 370 processors to handle this function. However, studies indicated that the

"The system has to support 6,000 terminals in 1,000 branch offices. Peak transaction rates were expected to approach 100 transactions per second."

cost of the hardware would be quite expensive, perhaps as much as $20 million to $25 million.

The study team then considered the use of a multiple-minicomputer approach as an alternative. The philosophy behind the multiple-minicomputer approach was to segment the data base into manageable subsets and have each subset supported by a minicomputer. Transactions would be directed to the appropriate minicomputer for inquiry updating.

Impressive Dimensions

The dimensions of the system are quite impressive. The system has to support up to 6,000 terminals located in 1,000 branch offices throughout the state. Peak transaction rates were expected to approach 100 trans/sec. The data base would consist of upwards to 4 million savings accounts and more than 4 million demand deposit accounts. The physical data base is estimated at 4G to 5G bytes. The system had to provide continuous uptime during the working day. The average transaction must be serviced within 3 sec and the 95 percentile of response time must be less than 6 sec.

All studies indicated that the limiting factor in building this system would be the rate of updates for the very large data base involved. CPU time was not considered to be a major factor because the transactions are relatively simple from a computational point of view. The only demanding part of the transaction would be the formatting of responses for the terminals.

An analysis of the file-handling system indicated that a moving-head disk would be capable of supporting approximately 8 trans/sec. The minicomputer approach would therefore consist of several computing modules, each operating a 8 trans/sec. Because the load was expected to approach 100 trans/sec it was concluded that a 12-module system would be required.

Each module would contain a unique segment of the data base. To this module would be attached a portion of the terminal base located in geographically contiguous branch offices.

It was expected that a customer would transact most business at a single location. However, it was also recognized that there would be exceptions to this rule; on occasion, a customer will be in a different part of the city or the state and want to access his accounts.

It is also possible that a customer may move. Therefore, a high-speed intermodule communications link was specified. Transactions arriving at the "wrong" module could be quickly shunted to the appropriate module for processing. The return message could be brought back to the original module and then distributed to the requesting terminal.

There are three types of modules. The Demand Deposit Account/Savings Deposit Account (DDA/SDA) module is the basic workhorse of the system. It processes inquiries to the data base and logs account updates.

The On-Line Data Capture (ODC) module is used to process transactions associated with the opening and closing of new accounts, changes of address, etc. These transactions are transmitted by the teller terminals connected to a DDA/SDA module and passed on the ODC module.

In addition, there is a Network Operations Control (NOC) module, which provides a facility for operator control, monitoring and statistics gathering. Critical NOC control functions are duplicated in other modules in case the NOC module fails.

The system is divided into two computing centers, one in San Francisco and the other in Los Angeles. These locations were chosen to be co-resident with the existing IBM 370 batch processing system. Communications is provided between the centers to account for transactions originating in one center that must be processed at the other.

The intercenter communications link operates at 9,600 bit/sec. Communications between modules in a center operates under IBM's Synchronous Data Link Control (SDLC) protocol at 1.2M byte/sec. The terminals communicate

RESOURCE MANAGEMENT SYSTEMS

RMS is a Software Solution for the IBM SYSTEM/34.

MANUFACTURING

- Inventory Control
- Bills of Materials
- Material Requirements Planning
- Production Control
- Capacity Planning
- Job Costing

DISTRIBUTION

- Order Entry
- Billing
- Purchasing
- Inventory Control
- Sales Analysis
- Forecasting

FINANCIAL

- Accounts Receivable
- Payroll
- Accounts Payable
- General Ledger
- Financial Budgeting

These integrated on-line systems have very modest hardware requirements. Migration to the System/38 is assured by original Data Base Design techniques. RMS can operate "as is" or be customized to meet your specifications by our staff of consultants and programmers. RMS is fully backed by extensive documentation, customer education and installation support.

For more information contact :

Professional Computer Resources, Inc.
(800) 323-3331
IN ILLINOIS CALL: (312) 932-2200
2021 Midwest Road Oak Brook, Illinois 60521
Offices: Chicago • San Francisco

"The cost of coding 115,000 lines of code was 50% less than that usually encountered in mainframe systems."

at a maximum speed of 2,400 bit/sec.

A module consists of two message-handling processors (MHP) and two file management transaction processors (FMTP). The processors work in pairs, with the load shared between them.

In the event of failure, one processor of a pair can carry the complete load. Switchover to work around a failed processor is done automatically. The computers from the original implementation of the system are 128K-byte General Automation, Inc. GA-16/440 minicomputers.

DDA/SDA processing is distributed between the MHP and the FMTP. The MHP provides all communications for the module. It provides line control, task scheduling control, message handling and routing functions.

The MHP determines if a transaction received on one of its communications lines is destined for its local data base. If not, the MHP will route that message to the appropriate module.

The MHP contains software for logging message traffic to magnetic tape. This log is used for recovery purposes and for off-line data processing to determine transactions statistics.

The MHP also supports a package for operator communications to the system and a capability to control the switchover of communication lines in case of processor failure. It also provides a data base load and unload package, transaction processing support routines, an operator communications package and miscellaneous utility routines.

Inherently Reliable

The system is inherently reliable. It is highly unlikely that any module could be lost to system operation, since no single computer failure can affect module operation. A disk failure would, at most, slow down module operation.

Even in the unlikely event that a module did fail, it would bring down only one-twelfth of the system, and this only for the time it takes to repair or replace the module. In an extreme emergency, the NOC module can be reconfigured to support a DDA/SDA role by switching lines to the NOC and by replacing disk packs in the NOC with the latest disk packs of the failed module.

Software is especially tailored to the application and consists of a relatively simple operating system software. Hence the software is expected to be highly reliable and should provide no major dc wntime to the system.

The system appears to have met its performance and cost goals. A vice-president of the bank has been quoted as saying, "In the cost area, the minicomputer hardware cost per transaction per second is about $50,000. This compares with about $300,000 in hardware costs per transaction per second for our former large-scale mainframe solution designed to meet similar uptime and reliability criteria.

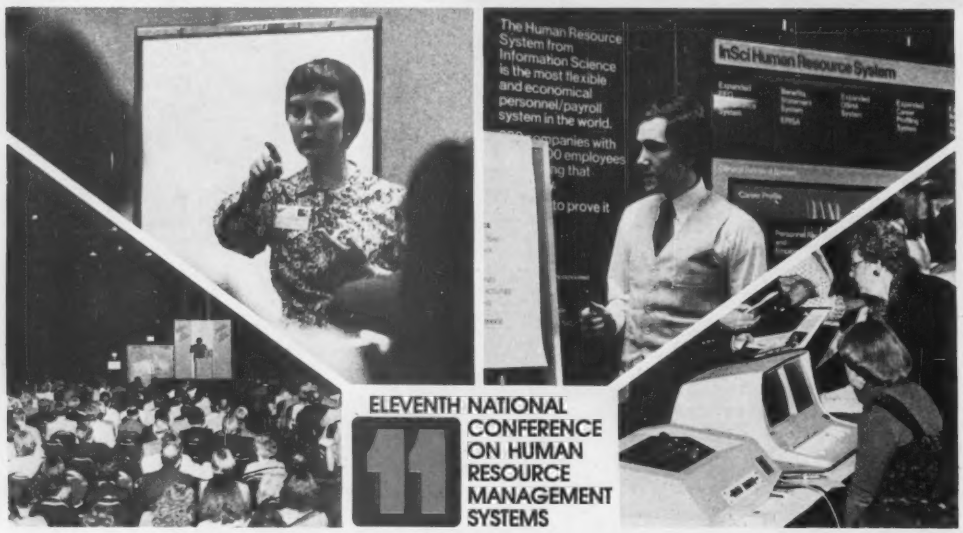
"In addition, our inherently modular design leads to easy low-cost system expansion with complete independence from geography or the size of the data base."

This same official stated that the cost of coding the more than 115,000 lines of code needed to make the system operational was about 50% less than that usually encountered in mainframe systems. He also claimed that because of the nature of the system, the communications network line cost was reduced to about 80% of what it would have cost with an equivalent mainframe system [11].

This system, when totally implemented, will consist of 14 modules containing 56 minicomputers capable of expansion to handle new functions. The limiting factor on expansion is the information carrying capacity of the intramodule SDLC links. These links can be increased in speed or in number to accommodate an expanded work load.

On-Line Order Entry System

This case study describes an on-line order entry and inventory control system built for a major manufacturer. The company is 100 years old and a leader in its field. It manufactures five major product lines, including 20,000


The Human Resource System from Information Science is the most flexible and economical personnel/payroll system in the world.
InSci Human Resource System

ELEVENTH NATIONAL CONFERENCE ON HUMAN RESOURCE MANAGEMENT SYSTEMS

Focus on the bottom line impact of human resource management.

In New York October 20-22.

If you're a data processing, personnel, payroll or senior management executive, don't miss InSci's Eleventh National Conference on Human Resource Management Systems – at the Grand Hyatt Hotel in New York City October 20-22.

The theme is "The role of personnel in the 1980's – a key to profitability." Each general session speaker will address the impact of personnel on productivity and the bottom line. Over 40 workshops will stress efficiency in human resource management and the critical role of computerized personnel and payroll systems.

There'll be detailed sessions on software design and installation, state-of-the-art systems demonstrations – plus unlimited opportunities to talk things over, and exchange ideas, with leading professionals and executives from coast-to-coast.

This is the annual event in human resource management systems. Last year's conference drew over 700 professionals. This year's will fill up early. For additional information, mail the coupon or contact Ms. Sheila Alper at 201-391-1600 today!

Information Science Incorporated
95 Chestnut Ridge Road
Montvale, New Jersey 07645

Mail to: Sheila Alper or phone 201-391-1600

Name _____
Title _____
Company _____
Address _____
City _____ State ____ Zip ____
Telephone _____

InSci
Innovators in people management

Microsoft is pleased to announce there will be no 16-bit software crisis.

They didn't realize it at the time, but when Bell Laboratories developed the UNIX* Operating System they solved some mighty weighty problems for 16-bit software development. Now Microsoft is picking up where Bell left off and putting the UNIX Version 7 OS on the Intel 8086, Zilog Z8000 and Motorola 68000. (We've got the PDP-11† version too, at a very reasonable price.) We call it the XENIX™ operating system, pronounced "zeenix." Naturally, we are customizing and enhancing the XENIX OS to meet your needs, and we are providing complete support for every customer.

That's great news for the legions of UNIX OS fans, as well as for everyone getting into the 16-bit market. But even better news is that Microsoft can offer XENIX to OEMs at very competitive prices.

16-bit micros and the UNIX OS—that's a powerhouse combination, and only Microsoft has it.

Finally, a state-of-the-art, standard OS.

As the biggest name in microcomputer system software, Microsoft will define the common ground to unite the next decade's microcomputer hardware.

And the XENIX operating system should make life a lot easier for everyone in the business. As the standard operating system, it will eliminate the crises that normally accompany emerging technology: the relearning and rewriting needed for each new machine. The software investments that become outdated the minute hardware diversifies. The wasted efforts.

The XENIX OS (written in the C programming language) will provide long sought-after hardware independence and portability, thus protecting software investments across hardware lines. Switching to a new microprocessor will be easy, since Microsoft will support them all.

Honest-to-goodness UNIX OS. Only better.

The UNIX system well deserves the attention it's received in the past decade. It's a highly sophisticated, interactive, multi-user, multi-tasking system with extensive utilities and accompanying software packages—creating a total working environment. A standard environment.

Actually, the operating system itself is only a small part of the XENIX package. The vast system of utilities, developed over the last 10 years at Bell Laboratories, includes a C compiler, software development tools, function libraries, games, text formatting and typesetting programs, and much more.

And Microsoft's modifications and enhancements make the XENIX OS even more suitable for general commercial applications. XENIX software will fully utilize the powerful instruction sets and large addressing capability of the 16-bit microprocessors.

As with all Microsoft products, the XENIX system will be customized to your exact needs and specifications, then supported, maintained and updated every step of the way.

In addition, all of Microsoft's already-famous system software (including BASIC, COBOL, Pascal, DBMS), will run on the XENIX operating system. XENIX will also run all existing UNIX Version 7 OS software.

A proven leader in worldwide software standards.

Microsoft's role as the leading supplier and authors of microcomputer system software has meant that we've established a number of standards throughout the industry.

That's why we're called a leader.

The establishment of a 16-bit standard operating system will be a big step forward for the industry.

With the introduction of the XENIX operating system, we're offering a superior standard system, plus the benefit or our extensive knowledge and expertise.

The XENIX operating system. The standard that ends the 16-bit software crisis. Before it's even begun.

*UNIX is a trademark of Bell Laboratories
†PDP-11 is a trademark of Digital Equipment Corporation

We set the standard.

10800 NE Eighth, Suite 819
Bellevue, Washington 98004
206-455-8080
Telex 32894

MICROSOFT

IDMS...The DBMS for On-Line Systems.

Leading database consultants have stated that it is an absolute must that any organization considering on-line systems use a database management system. However, on-line systems demand a particular type of database management system, one that is capable of providing unusually high performance plus elegant backup and recovery facilities. However, users are also very interested in tools for fast application development and easy access to data by end users in an on-line environment. IDMS provides these facilities.

As a result, IDMS has proven an indispensable tool in supporting on-line systems with an extensive number of the most sensitive and important applications for business, industry, and government. From this unique experience has emerged new versions of IDMS and related facilities to support on-line systems meeting the diverse needs of application developers, end users, and processing departments. IDMS, supported by Cullinane Corporation, has been particularly successful because it meets the six key factors for success in database software. They are:

1. Vendor Record: Cullinane Corporation has an unparalleled record for vendor support. (Number One every year in Datapro ratings.) And with an unequaled record in system enhancement and new product development.

Cullinane Corporation is also the most financially successful database software company with the best growth record and highest earning rate of all software companies. As a result, in 1978, Cullinane Corporation was the first computer software company in ten years to have a public offering, thus providing substantial financial resources for product enhancement, new product development, and company operations.

2. Control Via an Integrated Dictionary: The Integrated Data Dictionary serves as the control facility for all information about the database and data communications environment. In addition, it integrates all components of IDMS facilitating easy application development, end user access to data, and efficient processing in an on-line environment.

3. Facilities for Fast Application Development: IDMS is the most powerful DBMS in the world. It allows applications designers to truly mirror a company's operations within the limits of the company's computer processing capabilities. However, users also want to develop applications faster with less skilled people. Simply stated, IDMS has easy-to-use tools for input processing, data management, and output processing. Traditional programming is greatly reduced via the Application Development System, On-Line Query, CULPRIT Report Generator, and the INTERACT on-line program development and word processing facility.

4. Easy Access to Data by End Users: In addition to the user facilities described above IDMS has On-Line English, the only true English language query facility for top management and other users. Management needs to tap the application's data flow easily; but more importantly, it needs the information immediately. On-Line English makes this a reality. Also, the user will not need to know anything about the organization of the database.

5. Computer Processing Efficiency: While users of on-line systems like ease of use, they demand high performance and sophisticated backup and recovery facilities. IDMS was designed to meet these requirements through its architecture which fully integrates the database and data communications facilities. IDMS DB/DC is fully multi-threaded providing for efficient processing in both the update and retrieval modes. Also, an application developed under IDMS DB/DC using the Universal Communications Facility will run under any other TP monitor without modification. In addition, IDMS includes a Shared Database System which supports multiple CPU's sharing databases, eliminating problems associated with shared DASD.

With the introduction of the IBM 4300, users want a system capable of supporting distributed database networks. IDMS is the only database system architecturally designed to support not only distributed access to a database but distributed databases, and combinations thereof.

A facility important to the IMS/DL1 user is ESCAPE which allows use of IDMS with IMS/DL1 applications without modifying the application programs.

6. Protection: DBMS users want two types of protection—protection that their investment in database applications will be protected from any changes in IBM hardware or software. IDMS is written in such a way as to make it virtually impossible for IBM to make any practical move that would make IDMS-based applications obsolete. In addition, Cullinane Corporation contracts state that it has the obligation to keep each user current with any new IBM developments.

DBMS users also want protection that their IDMS applications can be audited by themselves or their respective audit firms. Cullinane Corporation is the world's leader in EDP audit technology and offers user-oriented audit software including a library of audit routines especially tailored to the IDMS on-line environment. Thus, IDMS provides the user with complete protection, the only DBMS to do so.

Attend a free seminar in your area on the DBMS for On-Line Systems.

Free Seminars

Tailored to management, these seminars introduce the complete family of integrated software systems from Cullinane. Particular emphasis will be placed on the six key factors to success in database systems and on IDMS, the DBMS for On-Line Systems.

Cullinane Corporation

Dates and Cities

SEPTEMBER
23 Omaha, NB
23 San Diego, CA
24 Greenville, SC
24 Newark, NJ
24 St. Louis, MO
24 Toronto, ONT
25 Houston, TX
25 Manchester, NH
25 Rochester, NY
30 Denver, CO
30 Indianapolis, IN
30 Jacksonville, FL

OCTOBER
2 Edmonton, ALT
2 Ft. Lauderdale, FL
2 Toledo, OH
7 Allentown, PA
7 Columbia, SC
7 Phoenix, AZ
7 Portland, OR
7 Vancouver, BC
9 Albuquerque, NM
9 Darien, CT
9 Jackson, MS
9 Madison, WI
9 Milwaukee, WI
14 Austin, TX
14 Moncton, NB
14 Nashville, TN
14 Oklahoma City, OK
14 Piscataway, NJ
15 Cincinnati, OH
16 Providence, RI
16 White Plains, NY
28 Baltimore, MD
28 Cleveland, OH
28 Columbus, OH
28 Grand Rapids, MI
28 Newport Beach, CA
28 Syracuse, NY
29 Montreal, QUE

NOVEMBER
4 Saskatoon, SASK
5 Albany, NY
5 Southfield, MI
6 Green Bay, WI
6 Salt Lake City, UT
6 Springfield, MA
6 Westbury, NY
12 Des Moines, IO
12 Ft. Wayne, IN
12 Poughkeepsie, NY
12 Seattle, WA
12 Tulsa, OK
13 Charleston, WV
13 New Orleans, LA
13 Saddlebrook, NJ
13 Tampa, FL
13 Winnipeg, MAN
18 Pittsburgh, PA
18 Roanoke, VA
19 Akron, OH
19 San Juan, PR
9 Minneapolis, MN
9 San Francisco, CA
10 Quebec City, QUE
10 St. Louis, MO
11 New York, NY
11 Richmond, VA
16 Boston, MA
16 Detroit, MI
20 Washington, DC
25 Sioux Falls, SD
26 Toronto, ONT

DECEMBER
2 Calgary, ABT
2 Chicago, IL
2 Memphis, TN
4 Charlotte, NC
9 Buffalo, NY
9 Cleveland, OH
9 Harrisburg, PA
9 Louisville, KY

I'd like to attend your seminar on _____ Date

in _____ City

My computer is _____

Name/Title _____

Name/Title _____

Company/Department _____

Address _____

City _____

State _____ Zip _____

Phone ( ) _____

Mail to: Rosalie Cope, Cullinane Corporation
20 William Street, Wellesley, MA 02181
Phone (617) 237-6600

Database: Cullinane

"The distributed approach has the advantage that no single failure could bring down the whole system."

separate numbered products.

Over the years, a very complex discount policy has evolved. Five years ago, the company used a 25-year-old order entry system based on Flexiwriters, paper tapes and an IBM 1050. Its successful operation depended on the knowledge of key people.

Orders were entered manually through the Flexiwriter and sent over communications lines to generate a paper tape at a central site. The central computing facility was and is an IBM 370/158.

The company had a strong policy of centralized computing. Each user of the central processor was charged back a proportional cost. It was in this set of circumstances that the company considered the use of automation to lower distribution costs.

The distribution network was spread over 12 geographical locations. The process was very labor-intensive, with a high error rate.

A management consulting team was brought in to survey the problem. To reduce costs, the consulting team suggested that the company reorganize into only four distribution points. Each distribution point would be responsible for order entry, inventory control and the control of associated warehouses.

Because the reduction in distribution points would increase the work load at each point, it was determined that an automated system would be necessary for adequate service. This system would provide the capability for computer-controlled order entry, for the processing of orders on-line and for on-line inquiries against order and customer files to determine the status of an order. It would also provide an on-line inventory system.

The introduction of such a system would save $700,000 a year in distribution costs. Against this would have to be weighed the cost of automation.

Two Alternatives

Two approaches were considered for providing automation at the distribution point. The first was to upgrade the existing mainframe so that it could provide on-line teleprocessing support. Terminals supplied by the manufacturer would be located at the four distribution points. These terminals would be connected to the central computer via dedicated communications lines.

The second alternative was to locate minicomputers at the four sites. Each minicomputer would support a local data base comprised of orders, customer files and inventory appropriate to that site. The minicomputer would be configured so that all orders could be processed locally.

A trade-off study was initiated to determine which of these two approaches would be most advantageous to the company. An estimate of the cost for each of the two approaches is shown in Figure 2. These costs are amortized over a four-year period.

We note from the figure that the minicomputer approach was evaluated to be slightly less expensive than the mainframe approach. This advantage was not deemed sufficient in itself to determine the use of the distributed minicomputer approach.

A service comparison, however, indicated that there were significant advantages to the use of minicomputers. These advantages were perceived by the distribution people — users of the system — to be overwhelming, and it was for these reasons that the distributed approach was chosen.

Three items were considered: response time, system reliability and risk. The central computing people could not guarantee or even estimate what the response time of the system would be.

With the central approach, the entire system would be dependent upon the availability of the central computer. History has shown this availability to be approximately 92%. The distributed approach has the advantage that no single failure could bring down the whole system. The user might lose a branch office or the central data base backup, but not the whole system.

The central computing people could not guarantee the delivery time or a

WHERE WOULD YOUR HARDWARE BE WITHOUT SOFTWARE?

Your hardware would be in the dumps without exceptional system software to guarantee maximum productivity.

Nixdorf Computer Software Company (NCSC) is a world leader in system software for users of IBM System/360, 370, 4300 and plug-compatible computers.

The many software products NCSC has designed to improve your installation's productivity include EDOS/VS—an operating system with increased productivity over IBM DOS/VS and DOS/VSE; PDMS—an online programming system; ECON—an enhanced console system with interactive facilities; NCSC Database and NCSC Datadictionary—an efficient data base management system; and DEX—an online data entry system.

Productivity through better software is what NCSC is all about. You owe it to yourself to keep your hardware out of the dumps and running at maximum efficiency. Contact any one of our offices to obtain more information and to schedule an installation in your operation.

NIXDORF COMPUTER SOFTWARE COMPANY

Richmond 804/276-9200, Los Angeles 714/634-0733, Chicago 312/635-6310, Dallas 214/247-2004.

NIXDORF COMPUTER

WHEN YOUR COMPUTER'S DOWN, ARE YOU OUT OF BUSINESS?

You need the Tandem NonStop™ System. It will keep on running right through a failure which would shut down any other system on the market today.

And never lose or duplicate any transaction in process at the time of a failure.

Providing absolutely unsurpassed protection for the data base.

Expanding with all its original software. Even the operating system. Growth with supplements, not replacements. Unusual.

Expandable without penalty from the basic two processor system to a four thousand processor system which blankets the globe.

Transaction oriented, with an extremely low cost-per-transaction. Cost efficient. High throughput. Wherever computer downtime equals irreversable or non-supportable loss. Dollar for dollar, it's the soundest investment you can make for high volume transaction processing.

TANDEM

Tandem Computers, Inc.
Toll Free 800-538-3107 or in California (408) 725-6000
Branch Offices throughout the U.S.A., Canada, U.K. and Europe. Service throughout the world.

Here's the answer.

A Tandem NonStop System. From $150 thousand to $3 million. Expandable without cost or performance penalties to sixteen processors per system and up to 255 systems per network. Local or global, no system of comparable cost can even approach the performance.

Tandem Computers, Inc.
19333 Vallco Parkway, Cupertino, California 95014
Attn: Marketing Communications, Department 109

☐ Please send me your introductory brochure on the benefits and applications of NonStop computing.
☐ Please arrange a demonstration of your unique NonStop capabilities.

My potential application is in _____

Name _____ Title _____

Company _____ Street _____

City/State/Zip _____ Phone _____

"Distributed processing is just another tool. It is not a panacea or, as some would maintain, evil incarnate."

development cost for the system. However, several turnkey vendors were willing to guarantee the costs, delivery time and performance of a minicomputer approach.

It was based on these guarantees that the distribution people and the central computer people jointly agreed that a procurement would be held for a distributed minicomputer approach.

This system encompasses four sites, each with a minicomputer supporting from four to 16 terminals. The minicomputer chosen was the Data General Corp. Nova 830, with 64K words of main memory. The system supports dual data base recording for reliability on the disk.

Several terminals are local to the computer and support order entry and entry functions. Other terminals are remotely located in warehouses and provide inquiry and printer functions for the warehouses.

Each system is completely self-standing and can operate without support from the mainframe. The operating system is DG's RDOS. All support is standard except for the dual data base recording, which was specially provided for reliability.

A standard package provided by DG is used at night to make the computer system look like an IBM 2780 terminal to the host IBM machines. Summary files are shipped to and from the main site by means of a dial-up line.

The system went on the air in 1975 and has been upgraded once to accommodate larger data bases at some of the sites. The system has proven to be reliable and to provide adequate response time.

This last case history describes a study performed for a county government in the Washington, D.C., area. This government provides a central computing facility for its user community. The computer was an IBM 370/158-II.

In this system environment, all computing and programming support for the county is centralized in the computing center. This center supports approximately 15 different on-line operations concurrently with batch. The on-line operations run during the first two shifts of operation; batch runs all day long. It is a Cobol shop.

Significant growth was anticipated, creating the need for a system upgrade.

Several solutions were considered by the county. One was to add another IBM 370 to handle the excess work load. A second solution was to replace the existing 370/158 with a more powerful, IBM-compatible mainframe. The third solution was to augment the existing mainframe with a set of minicomputers that would physically coreside with the mainframe. This approach would require conversion of certain programs to run on the minicomputer.

The advantages identified for the minicomputer approach included the potential for lower and more predictable response time, the use of hardware redundancy to back up critical applications and the ease of introducing lower cost incremental upgrades.

Those advantages were weighed against several disadvantages. First, it was determined that individual Cobol programs would run slower on the minicomputers than on the mainframe. This might negate the response time advantage.

Second, new applications might require hardware procurement, since the minis provided only limited computing power. Considering the procurement environment within the county, this could introduce severe delays.

It was determined that there would be a technical risk and potential disruption associated with converting existing applications to the minicomputers. Some retraining of program staff would be required. The computing center would probably have to work with multiple vendors. More operating staff would be required. More floor space would be needed and the minicomputer upgrade would provide less total computing power than a replacement of the mainframe by an IBM 3031 or equivalent. Moreover, the proposed minicomputer configuration would include no networking; any in-

The deal of 1980!

Computer SystemCenter

CP/M SOFTWARE: ACCOUNTS RECEIVABLE
ACCOUNTS PAYABLE
GENERAL LEDGER
PAYROLL

ONE TIME OFFER: For $500.00, TCS Corporation's Microsoft Basic, CP/M compatible software sounds like a bargain. But, wait until you hear about it's many disadvantages!

A LITTLE HISTORY:

For the past four years, TCS Corporation has been selling Pertec (MITS) business systems as one of their largest and oldest dealers. Since we were so early in the microcomputer marketplace, we were forced to develop an extensive array of business software. Some of this software has been marketed for years by Pertec.

During 1978, a set of four business systems was developed for sale in the CP/M environment under the name "Peachtree Software". However, during this period there was a split among the principals which resulted in a parting of the ways – TCS Corporation and Retail Sciences (RSI).

It was decided that RSI would retain the "Peachtree Software" trademark and for a period of one year, TCS would be restricted from competing with them. So, for the past year, RSI has been enhancing and marketing their version, and now TCS is ready to market our version to you.

EXACTLY WHAT IS IT?

A fully integrated business software arsenal including:

General Ledger - provides immediate financial information for your company by keeping thorough record of all financial transactions.
Accounts Receivable - instant customer account information (current and aged) with complete invoicing and statement capabilities.
Accounts Payable - maintains complete vendor/voucher history including check writing capabilities.
Payroll - calculates payroll for every type employee while maintaining monthly, quarterly and yearly totals for reporting purposes in multiple states. User modifiable tax tables. W-2, 941's, etc.

TALK OF VALUE!

This offer is absolutely the best software value we have ever heard of. For only a one-time charge of $500.00 you will receive all four systems to use at your discretion. They will form a solid base for your marketing efforts as they are or with your enhancements.

TCS CORPORATION
P.O. Box 47980, Atlanta, Georgia 30362 / 404-455-6162

SYSTEM REQUIREMENTS
- 8080 or Z80 CPU
- 48K RAM
- Microsoft Basic, CP/M

SUCH AS
- Mod II
- Tandy II
- Cromemco
- Dynabyte
- Altos
- Apple
- North Star
- Heath
- Super Brain
- Ohio Scientific
- PCC 2000
etc., etc. . .

OFFER INCLUDES
- all four systems
- source code
- sample data
- comprehensive users manuals

Ask for Lynn . . . she will be waiting to hear from you.

If you haven't received this letter, call us . . . 404-455-6162

terchange between machines would be done by manual carrying of magnetic tapes.

Study conclusions were as follows. The qualitative advantages and disadvantages were determined to balance out. The only inherent advantage of the distributed approach was reliability since a backup processor was provided.

However, reliability was not an issue with the users. The IBM 370/158 provided significant uptime and support for its customers.

Because there were no inherent advantages to either approach, it was determined that the choice would be determined by cost. If the costs were equal, it was recommended that the mainframe upgrade be chosen since it was potentially less disruptive and provided more computing power suited to the county's needs.

An estimate of the costs of both approaches is shown in Figure 3. It can be seen that the distributed approach over a five-year period would cost $100,000 more than the mainframe approach. Significant costs were associated with software conversion.

Since the cost benefits slightly favored the mainframe and since the mainframe upgrade was the least disruptive approach, it was concluded that the county should upgrade its mainframe. Based on this study conclusion, the county contracted for a mainframe upgrade.

Summing Up

We have identified two major categories of system applications. The first category requires a dedicated facility because of high throughput and low response time requirements. We call these Category A systems. The second type, called Category B systems, are more modest and could exist as part of a shared user system.

Both categories are amenable to either distributed or mainframe solutions. Each solution has its pros and cons. We conclude that each case must be judged on its merits.

It is perhaps less impactive to consider distributed processing for Category A applications since by definition they require the establishment of a dedicated computing center. Certainly the management issues are not as complex as in Category B systems.

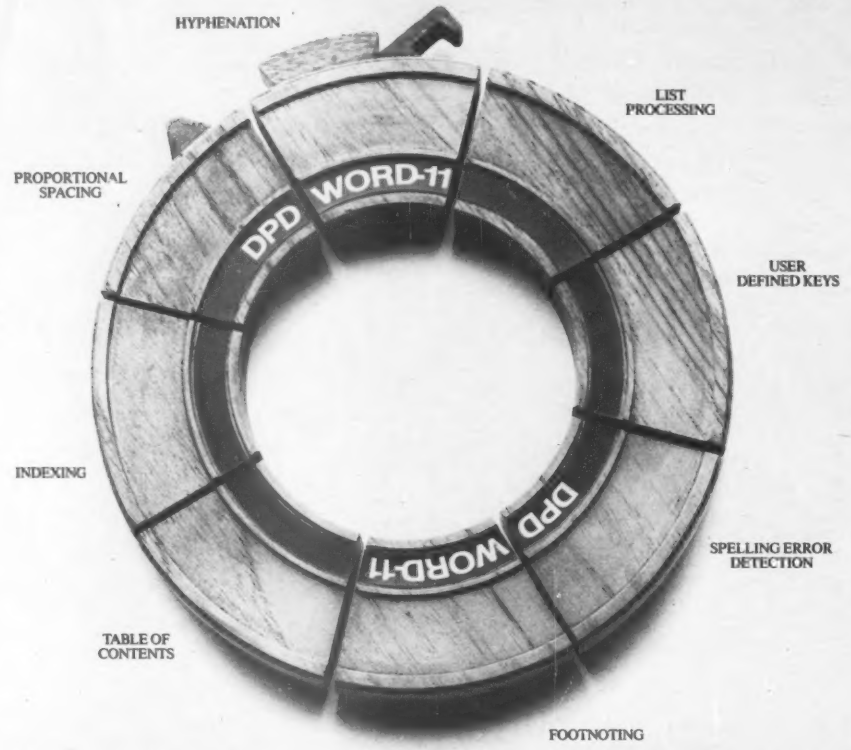
To make the evaluation, a cost-effectiveness analysis should be made. Definitive evaluation criteria should be established. To the extent possible, quantitative measures should be used.

Above all, it should be kept in mind that distributed DDP processing is just another tool. It is not a panacea or, as some would maintain, evil incarnate. It should be used where appropriate to the organization and task at hand. □

FOOTNOTES

1. Liebowitz, Burt H. and Carson, John H. "Tutorial: Distributed Processing," 2nd ed. (IEEE Computer Society IEEE EHO 127-1, 1978, Pg.1).
2. Wirshing, J. "Computer of the 1980's — Is It A Network of Minicomputers?" *Proceedings of Compcon (Fall 1975)*, Pg. 23-26.
3. Fuller, S. "Price/Performance Comparison of C.mmp and the PDP-10." *Third Annual Symposium on Computer Architecture* (1976), Pg. 198-202.
4. McGill, R. and Steinhoff, J. "A Multi-Microprocessor Approach to Numerical Analysis: An Application to Gaming Problems." *Third Annual Symposium on Computer Architecture* (1976), Pg. 46-51.
5. Connell, H. "A Minicomputer Network for Optimal Moving Target Indication." *Proceedings of Compcon (Fall 1976)*, Pg. 233-236.
6. Rosenblatt, R. "How GI Captured The OTB Stakes." *Electronics* (March 1, 1973), Pg. 66-67.
7. General Automation Press Release, Anaheim, Calif. (Feb. 14, 1977).
8. Schwartz, Mischa; Boorstyn, Robert R.; and Pickholtz, Raymond L. "Terminal-Oriented Computer-Communication Networks." *Proceedings of the IEEE*, Vol. 60, No. 11 (November 1972), Pg. 1415-1418.
9. Henson, Carl C. and others. "NASDAQ Concentrator System Specification." International Computing Co., Bethesda, Md. (May 18, 1979).
10. Foster, John D. "The Development of a Concept for Distributive Processing." *Compcon 76 Proceedings*, Twelfth IEEE Computer Society International Conference, 1976. Pg. 28-31.
11. Foster, John D., "Distributive Processing for Banking." *Datamation* (July 1976), Pg. 54-56.
12. General Automation Application Note, "Bank Distributed Computing Facility." (n.d.)

The Whole Truth About PDP-11 Word Processing.



When you explore word processing systems, you'll find a number of systems that offer part of the package. And frankly, if you're only looking for text editing, almost any system will do.

But if you're looking for full word processing capabilities, you should insist on a system that can handle all of the tough tasks you'll encounter.

For starters, a good system will have list processing, the vital function that generates correspondence, reports, even statistical analyses. It should have user defined keys to simplify repetitive operations.

And a good system can handle the details. Like spelling error detection, automatic renumbering of footnotes, tables of contents and indexes. And proportional spacing and hyphenation for professional looking documents.

A good system can be shared. It should be able to support up to fifty terminals. And it must be backed up by successful installations and a strong service team.

After you've examined the options, we think you'll select WORD-11, the only system with all the sophisticated features you could want.

For the details, please call Data Processing Design at (714) 993-4160. 181 W. Orangethorpe, Suite F, Placentia, CA 92670.

Data Processing Design, Inc.

AUTHORIZED DIGITAL COMPUTER DISTRIBUTOR

TSO CONTROL?

IT'S A PIECE OF CAKE

TSO is great because it allows programmers to be more creative and productive. But like many versatile capabilities, it must be carefully managed and controlled to guard against explosive resource consumption and related response problems.

This basic point has been recognized by over 360 installations world wide that control TSO with TSO/MON, the software product that tells you about your users, your system, and your management plan on a daily, monthly, and as-needed basis.

TSO/MON provides the data, the reports, and the methodology for:

- ***Daily Review of Service and Performance*** **- is response time on-target, when do problems occur?**
- ***Problem Tracking*** **- is it the user, is it the system?**
- ***Continued Management Evaluation*** **- what are the response, workload, and system availability trends?**

TSO control can be difficult or it can be easy. Send for our Evaluation Booklet and find out why, with TSO/MON, it can be a piece of cake.

mma
Morino Associates, Inc.
8133 Leesburg Pike, Suite 602
Vienna, Virginia 22180
(703) 734-9494
TELEX NO. 248659

Please send me the TSO/MON Evaluation Booklet:

Name _____

Title _____

Company _____

Address _____

City/State/Zip _____

Telephone _____

Morino Associates, Inc.
8133 Leesburg Pike • Suite 602
Vienna, Virginia 22180
(703) 734-9494

When it comes to

you're right on-line with us.

Cyborg Systems' highly regarded (so highly that there are more than 100 users to date) payroll/personnel system is now even *more* highly regarded.

Our new line is on-line.

And we now offer these time, labor and money saving features thanks to this powerful move forward into the 80's:

- Hard copy detail audit reports with change totals by company, to allow a complete record of all changes entered through the CRT.
- Set up of new, and maintenance to existing employees through the new hire form on and through CRT.
- Time data and labor distribution may be entered and balanced with on-line verification of the employee number, and editing of all the entered fields.
- CRT manual checks, adjustments and balances; immediate update of hires and terminations; on-line maintenance of skills, job history, education, etc.
- You create customer screens via the Cyborg report generator, allowing departments to tailor the screens around their needs ... without program modification.
- User has complete access to the file. Inquiries may be against any of the payroll fields: current, month, quarter and year to date dollars and hours. You may also inquire against all personnel data skills, education, salary benefits, etc.

Put your payroll/personnel problems in the Cyborg system ... and you'll see instantly that they're no longer problems. Contact us now. Get on the line so you can go on-line, or drop us a note today.

Cyborg Systems, Inc.

When it comes to people...people come to Cyborg.

21st Floor/2 North Riverside Plaza/Chicago, IL 60606/(312) 454-1865/In Canada 2742 Ross Avenue, Cornwall, Ontario K6K 1E5/(613) 938-7431/2091 Business Center Drive/Irvine, California 92715 (714) 955-0870

By G. T. Orwick

THE PLIGHT OF PROGRAMMING

Technological changes in the DP industry and the tremendous growth of the application software industry will have major impacts on the individual DP installation of the '80s.

We will examine these impacts in some statistical detail, but chief among them will be the migration of programmers into application software companies as these companies grow at a projected 30% annual rate. This exodus will leave even fewer and more expensive resources available for individual DP managers to perform other development activities.

Additional software development tools are in the works. They will be used as well as marketed by application software firms so that software development can be facilitated. But an overall increase in programmer productivity is not expected to make any major change in the resources required for system development.

The data processing industry is one of the fastest growing segments of the world's economy. In the span of approximately 25 years, the industry has developed to a state of adolescence, if not full maturity in some respects. Fortunes have been made (and lost) in computer manufacturing, peripherals, services, consulting, software and even the misuse of computers through fraud.

One of the most widespread complaints within the industry today is "I can't get enough good people to support my users' needs." As we travel around the world dealing with hardware vendors, software firms, consultants, DP executives and managers at all levels in all sizes of companies, the lack of qualified systems analysts and programmers seems to be a universal problem.

We will attempt to corroborate these feelings with the scanty statistical facts available in order to determine the severity of the problem and to track the trends which will continue into the future as the applications software industry begins to realize its full potential.

G.T. (Tim) Orwick is director of product planning at MSA, Inc., Atlanta, Ga.

The number of computers installed and the number of computer sites is growing at an increasing rate. Figure 1 shows the estimated computer population in thousands of machines for 10-year increments from 1955 through 1985.

These machines are broken down into general-purpose computers and dedicated machines which are typically used in process control and manufacturing operations. Currently, there are approximately 100,000 computer sites with a total of over 150,000 general-purpose computers installed.

One reason for the increasing rate of growth in the number of machines is the well known fact that computer costs are decreasing. The average hardware cost to perform 100,000 calculations has dropped from approximately $1.26 in the early 1950s to 12 cents in 1964 (with the advent of the IBM 360 series) to 1 cent in 1979 (with the large 370 machines) and on to a current price of about .05 cent (with the IBM 3033).

This trend is continuing with the recent price cuts on large IBM machines, as well as the announcement of the 4331 and 4341 equipment. The announcement next year of the rumored IBM series H equipment should again halve the average cost per calculation.

The number of programmers available to develop software for this ever-growing population of machines has not kept pace with the growth of machine population. Programmer population was estimated at 10,000 in 1955, 80,000 in 1965 and 220,000 in 1975. The estimated 1985 programmer population is 300,000. Figure 2 represents the growth in the programmer population and its trend for the user.

This discrepancy in the number of programmers per machine, which is shown in Figure 3, indicates that although we had 10 programmers per machine in 1955 with very little software available, we can project only 1.9 programmers per general-purpose computer in 1985. If dedicated machines are included, there will be less than one-third of a programmer for each machine installed in 1985.

Alarming Trend

This trend would not be alarming were it not for the fact that computer programming has not progressed beyond the stage of a "cottage industry." There have been no significant advances in computer programmer productivity thus far in the history of the industry. Current estimates place the growth of programmer productivity at approximately 3% per year, with no foreseeable change in this pattern.

The ever-increasing salaries paid to data processing professionals plus the relatively low increases in productivity mean the average design and programming cost per computer language instruction has risen from approximately $4 in 1953 to approximately $8 today. Thus, a fairly simple system with 20,000 instructions costs nearly $160,000 to design and program. A complex system containing over 300,000 lines of code would call for a current development cost of nearly $2.4 million.

User expenditures for applications software is a fairly difficult statistic to obtain. However, IBM estimated in 1975 that the total user investment in applications software for IBM equipment exceeded $50 billion. The estimate for applications software investment for all machines would be roughly twice that number. Current expenditures are estimated at approximately $80 billion for all software and is expected to increase to $139 billion by 1985.

DP Service Industry

We have shown that the hardware business is growing rapidly and that software is growing even more rapidly. In fact, the entire computer services industry — which includes facilities management, contract programming and consulting, applications and work systems software as well as data processing services — is growing at nearly 20% per year.

The number of companies in the DP

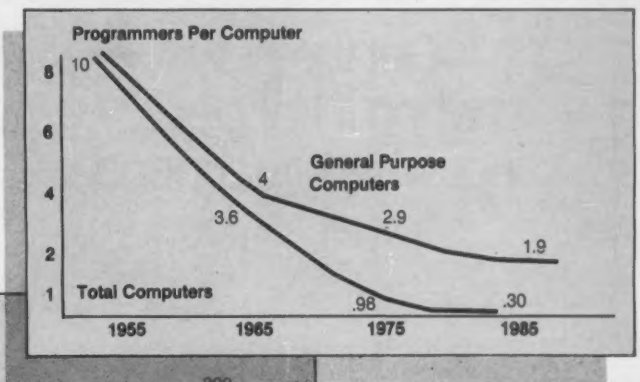


FIGURE 3

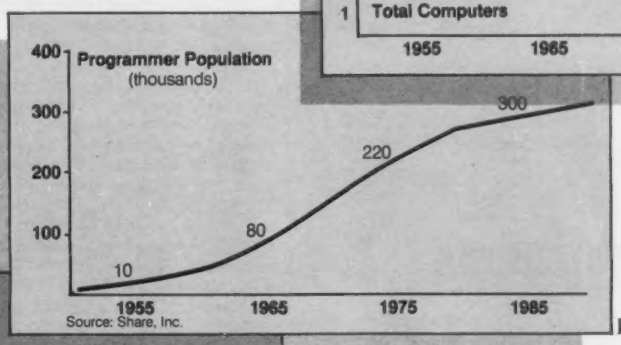


FIGURE 2

Computer Population
(thousands)

	1955	1965	1975	1985
General Purpose	1	20	150	925
Dedicated	0	2.5	220	330
Total	1	22.5	370	1255

Source: Share, Inc.

FIGURE 1

Low prices on the HP 3000. Great software too.

ASK.
We have the answers.

MANMAN® Family of Products.
Integrated manufacturing and financial management systems on the HP 3000.
Call for fall seminar dates in your city.

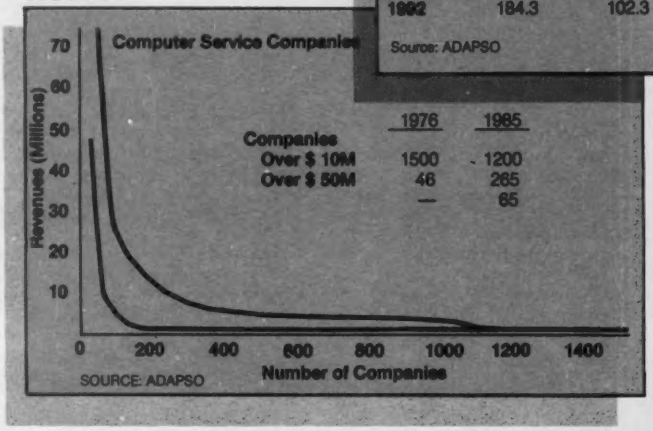
ASK Computer Systems, Inc.
730 Distel Drive, Los Altos, Ca. 94022
(415)969-4442

FIGURE 5

Application Software Industry Growth	30%	25%
1977	3.6	3.6
1980	7.9	7.0
1983	17.4	13.7
1986	38.2	26.8
1988	83.9	52.4
1992	184.3	102.3

Source: ADAPSO

FIGURE 4



services industry is expected to decrease through mergers and business failures for the foreseeable future. Figure 4 shows the relative size and number of companies in the services industry as determined from a 1976 study by the Association of Data Processing Service Organizations (Adapso).

Within services, the fastest growing segment, according to the Adapso survey, is the applications software industry. The study projected an annual growth rate averaging 30% through 1985 for applications software. Figure 5 shows the estimated size of the applications software industry in three-year increments from 1977 through 1992, assuming both a 30% and a more conservative 25% industry growth for that period.

The reasons underlying this growth are largely economic and managerial. A Share, Inc. study developed in 1976 stated that "current estimates are that lifetime maintenance of software is twice the initial development cost." The software industry offers an alternative to the individual DP manager through continued enhancement and maintenance of products.

Another major economic benefit of packaged applications is that more accurate implementation time schedules and costs can be estimated before the project is begun than can be obtained with a complete development project.

Because packaged applications are designed for ease of maintenance by the vendor, they also make it easier for the individual user to make changes, enhancements and modifications to suit his own needs.

It is our belief that as the number of programmers per computer decreases while each person's productivity shows little increase, an even smaller number of programmers will be available to the individual installation than was projected earlier.

Why will this happen? The growth of the applications software industry will demand so many programmers that there will be even fewer left for the individual installation.

Before exploring this further, we must point out that there were some unknown factors in the earlier projections. First, the growth of the hardware population was a fairly speculative number. Although we know it typically takes seven years from drawing board to practical application, thus precluding the possibility of any major shift in hardware technology before the late 1980s, we may be forced to change the definition of "computer." Many of the installations are particularly dedicated machines so they may not be representative of the type of installation normally thought of under the description of "computer" for purposes of measuring computer/programmer ratios.

Other factors, such as minicomputers, microprocessors and the shift toward communications networks, may alter the computer population figures through a redefinition of this sector.

Another great unknown which contains a great deal of subjective and qualitative aspects is the amount and type of software which is produced.

Get the news when it happens, not when it happens to get to you.

To subscribe, you can use your American Express, Visa or Mastercharge card and call, toll free, 800-343-6474 (in Massachusetts, call 617-879-0700).

COMPUTERWORLD
THE NEWSWEEKLY FOR THE COMPUTER COMMUNITY

ductivity and the effect of inflation multiplied to the original 30% annual growth estimate provided by the Adapso study.

The final column of Figure 6 shows the relative size of the application software industry compared with 1977. It indicates that the industry will be roughly five times as large in 1986, nine times as large in 1989 and over 15 times as large in 1992 as it was in 1977.

Since the application software industry is labor-intensive and since that labor is concentrated in the programming resource, it appears reasonable to assume that the industry will need three times as many programmers in 1983, five times in 1986, nine times in 1989 and 15 times as many programmers in 1992 as were employed in the industry in 1977.

Furthermore, because the industry and each company within the industry is absolutely dependent upon the production and continued maintenance of application software for their only products, these companies will be willing to pay whatever price is necessary to obtain these programmers on the open market.

In other words, because programmers are the application software company's "production line," the application software company will hire as many programmers as it needs to develop its products. If these programmers are expensive, the application software company will raise the price of its software on the market. And because the programmers are expensive, the DP community will buy the applications because it will not be able to hire enough programmers to develop in-house systems.

Figure 7 graphically depicts the projected total programmer employment and the proportion of those programmers who will be employed by application software firms. It appears that it will be a luxury (and probably a very expensive one) for an individual DP installation to have anything more than a maintenance-oriented staff by 1990.

Although Figure 7 is slightly biased toward application software firms, our projections show employment of nearly 125,000 programmers by 1985 (out of a total of 291,000). This contrasts with a total of 27,000 vs. a total population of 233,000 in 1976. The result is that the number of programmers available outside applications software firms will actually decrease over the next decade as application software firms continue to employ more programmers than enter the job market each year.

Other Major Trends

Application software firms are successful today because more and more organizations are turning to application packages. The major reason for this is that their current systems are obsolete because of changes in either accounting or DP technology; they lack resources internally to develop new systems themselves; the future direction of their hardware capabilities is often uncertain, particularly in light of the rapid changes in this technology; and, finally, management and internal user demands force the DP department into greater service levels without additional resources.

The 1976 Share study pointed out that future application trends will be led by three major items: the integration of all accounting and DP functions into sets of applications; the use of terminals for user communication directly with the application; and management involvement and demands upon the information systems of the organization.

The growth of the application software industry will provide viable alternatives in these areas as the industry matures and the marketplace grows. As a result, by the mid-1980s, there will be strong incentives for purchasing applications rather than developing them in-house.

In short, the decade of the '80s will be an exciting one for the data processing community as a whole. Naturally, we believe that our industry, application software, will remain at the forefront of growth and user involvement through this period. □

1. ASI/INQUIRY Is Easy To Learn and Easy To Use.

Because inquiries are stated in simple English, nonprogrammers can learn to use ASI/INQUIRY quickly. DL/1 structures are completely transparent to the user. You need not understand the complexities of multipathing or multiple data base access. Comprehensive diagnostic messages simplify error correction. ASI/INQUIRY automatically displays data in the appropriate format—horizontal, vertical, or overflow. Or you can specify any desired screen format. Repetitively executed queries can be saved in an on-line catalog. New release 5 features include the ability to defer query execution to batch and a powerful user exit facility.

2. ASI/INQUIRY Assures Faster Access and Response Time.

ASI/INQUIRY lets you access your DL/1 data bases through IMS or TSO faster and more efficiently. That's because it eliminates need to write and debug those highly procedural programs usually required to access data bases. ASI/INQUIRY operates as an IMS message processing program executed from any IMS DB/DC-supported terminal. Execution priority is dynamically controlled through automatic program message switching. High initial priority assignment assures fast response. Priority is then automatically adjusted to the rate that to-be-displayed data is encountered, which optimizes *load leveling* of IMS DB/DC resources.

3. ASI/INQUIRY Provides Complete Security.

Built-in safeguards protect data at the system, terminal, and data base levels. Data base administrators define the data bases users can access, their passwords, and the terminals from which individual data bases can be accessed.

Learn why ASI/INQUIRY is more heavily used than any competing product. Call or write—today!

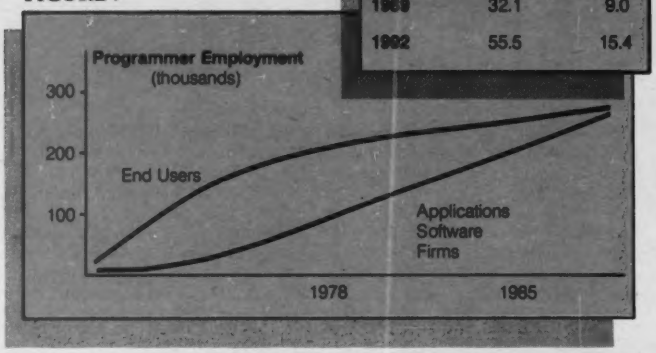
Applications Software, Inc.
21515 Hawthorne Boulevard
Torrance, CA 90503
(213) 540-0111

Member SIA Software Industry Association

FIGURE 6

Tempered Projections	20%	x1977
1977	3.6	1
1980	6.2	1.7
1983	10.7	3.0
1986	18.6	5.2
1989	32.1	9.0
1992	55.5	15.4

FIGURE 7



Moreover, the number of programmers and the growth of programmer training is also difficult to estimate. It may be that we will need to change the definition of "programmer." As our children play with microprocessors at home, we may find that a programmer in the traditional sense is no longer required for many of the tasks which we today assign to this class of professional.

Finally, the measurement of programmer productivity was not entirely objective. We seem to fall back on number of instructions produced, rather than attempting to define systems in any quantitative way that could be used to measure the overall effectiveness, rather than efficiency of a given programmer or group of programmers.

It is possible that some improvements can be made in all of these areas. For instance, a new programming language such as Pascal, which has been getting a great deal of notoriety lately, might improve programmer productivity to a great extent, although "high-level" languages in the past have not made any significant gains in this area.

We have read a great deal in the past few years about "software engineering," including structured design, modularization and the use of data bases. These techniques may be capable of extensions which will increase the total productivity of programmers and systems designers. In practical application, however, we find that the gains are not as high as their proponents would have us believe.

There are a number of design tradeoffs which can — and will — be made as hardware costs continue to decrease. It is really no longer necessary to write the most efficient program possible to perform a routine task. It is easier to trade off a little memory or a little disk storage to save time in the development and maintenance processes.

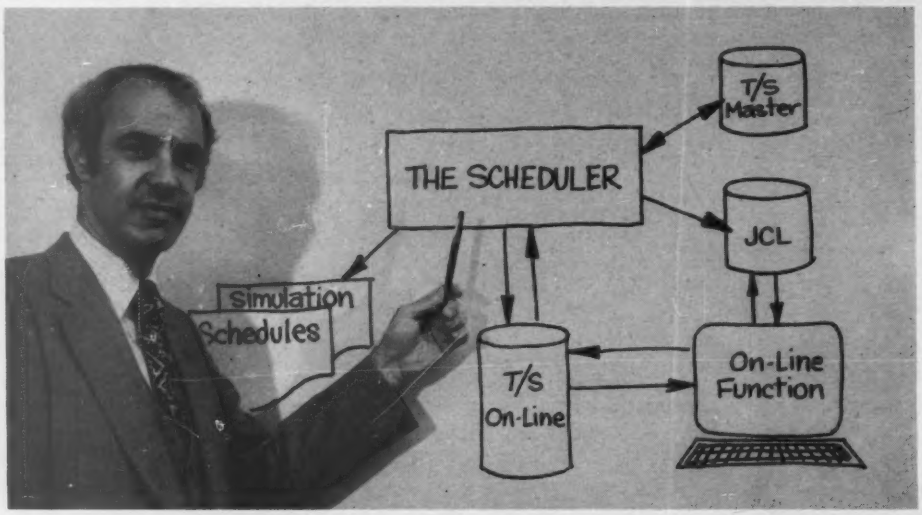
Another major area that is getting much consideration by software companies is software which itself generates application software. A growing number of development aids can be used to assist in the software development process. Some companies have developed their own "application generators" which can provide a fairly substantial core of code for routine accounting and clerical functions. These tools can increase productivity in the development of at least certain types of software.

Finally, user involvement in system design and development via terminals, query languages, report writers and the like will take much of the burden off the programmer in the future and place it on the non-technical user of DP services.

The Programmer Migration

Normal advances, as mentioned above, occur in programmer productivity, hardware technology and the like. The projection of programmer employment by application software firms through the 1980s indicates one major impact this industry's growth will have on all DP installations — mainly, major competition for programming resources.

Figure 6 is a more conservative projection of software industry growth than Figure 5. By nearly a 20% growth figure, we have factored in both a nominal increase in programmer pro-



"*THE SCHEDULER* is the most complete operations management system available. Let me show you why."

Mark Berg, Systematics International

Data Center Scheduling
THE SCHEDULER handles up to 99 work stations with a unique, English-like scheduling language that eliminates calendars.

Production Control Management
THE SCHEDULER displays JCL and control cards at a job set-up station, incorporates any changes, then submits and releases jobs at the proper times.

Workload-Resource Simulation
THE SCHEDULER produces predictive schedules for each work station, based on resource availability and variations in process time due to scheduling conditions.

On-Line Tracking
THE SCHEDULER provides job sequence enforcement, deadline prompting, next station notification, purge tracking, and a unique memo logging capability, under MCS, TSO, ROSCOE, TONE, etc.

On-Line Documentation
THE SCHEDULER maintains, prints, and displays on-line, free-form documentation organized by job, work station, and user-defined key.

Data Base and Reports
THE SCHEDULER data base contains descriptive and historical information about each production job. It reports on jobs, work stations, and schedules.

systematics international
10 Park Center
Sterling, Va. 22170
(703) 430-0550

☐ Please send me information about THE SCHEDULER

Name _____

Title _____

Company _____

Address _____

_____ Zip _____

Phone (____) _____ Operating System _____

systematics international
10 Park Center Sterling, Va. 22170 (703) 430-0550

AT HOME WITH JAMES MARTIN

(Continued from Page 29)

So you have data base query languages, and you'd like them to be simple enough for an end user to use without having a DP professional.

Query languages flow into report generators. You'd like a language here that gives you the capability to structure reports in any form you want, either alphanumeric reports or graphics reports.

Report generators tend to flow into graphics languages, of which we have a few examples but not nearly enough.

Then we've also got application development languages, like IBM's ADF, DMS and others. There are a large number of these from different software houses. I've added up the total number of high-level languages, by which I mean something that is an order of magnitude higher than Cobol. There are more than 100 of them, and many do not come from the mainframe industry but from software houses and the minicomputer industry.

There is no one such language with which you can do everything you want. Maybe there never will be. Therefore, you have to tailor the choice of language to the type of application, and this is the whole trick in getting them to work effectively.

Does this mean the days of Cobol are numbered?

No, I don't think so. Everyone wants to know if we are going to see the demise of the application programmer in the next few years. I think it's going to be exactly the opposite. There won't be *enough* application programmers; we'll be very short of application programmers everywhere. The Cobol programmers are going to hang in there!

Returning to automatic programming, you said recently, "It's the most important area in computing right now." How does it work and what kind of realistic productivity gains can we expect?

I am going to avoid using the term "automatic programming"; the phrase I think one ought to use is "application development without conventional programming."

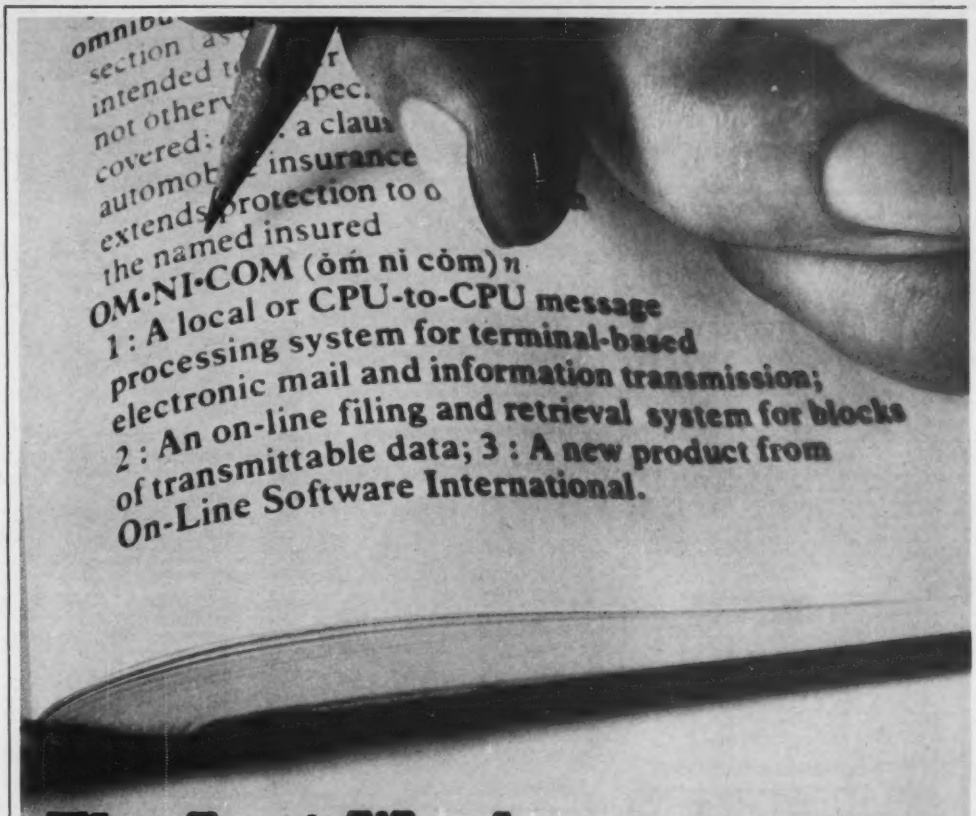
The languages work in many different ways. Some are report generators; some are query languages; some are application development facilities. But we can find very high improvements in productivity with these languages.

For example, in the generation of queries, it might have taken a Cobol programmer a week to create a query of a certain type for a user to employ at a screen. Now you can get the user doing it himself in 10 minutes.

Some corporations have measured the productivity which they've achieved with higher languages in terms of equivalent lines of Cobol — in other words, how many lines of Cobol *would* they have had to write *if* they had done it in Cobol. And some of them are quoting figures of more than 1,000 lines a day of equivalent lines of Cobol. The national average for Cobol programming is seven lines of code a day to 40 lines of code a day, varying with the skill of the programmer.

The difference between those two figures is so huge that we are forced to look at it very hard and ask, "Why hasn't that happened to everybody?" There are a lot of different answers to that question. A good part of the answers is that the high-level languages need improving — they're not yet good enough for everything one would like to do with them.

But that is understood by the mainframe vendors, the mini vendors, the micro vendors and the software houses. So I think we are going to see a


OM·NI·COM (ŏm ni cŏm) n
1 : A local or CPU-to-CPU message processing system for terminal-based electronic mail and information transmission; 2 : An on-line filing and retrieval system for blocks of transmittable data; 3 : A new product from On-Line Software International.

The Last Word in Electronic Message Processing Software

One of the most productive applications you can add to CICS has just been released by On-Line Software International. It's called OMNICOM, the electronic message processing system.

Following a simple, fill-in-the-blanks format, OMNICOM allows you to send a message to one person or many people on a "routing list." Chances are, you will receive an immediate response, but some people may prefer to hold their messages until a free moment. No problem. The computer reminds them that there are messages on hold. Either way, you know the status of your message…has it been received? Is it on hold?

More than for just interoffice memos, the OMNICOM system can be used between technicians for problem-solving…or, program the computer to generate messages while running specific applications. For any management need… inventory problems, exception problems, last-minute details… when the message needs to get across, you'll thank your computer for OMNICOM. Safe. Secure. Fast. And easy to use, with single or multiple CPUs.

Call today about OMNICOM, the load-and-go productivity system for your IBM computer.

ON LINE SOFTWARE INTERNATIONAL

65 E. Route 4, River Edge, NJ 07661
Phone: (201) 486-7770
800-526-0272

"The end user or the application developer languages become the tail that wags the whole data base dog."

huge improvement in the languages for application development without conventional programming.

In the same vein, you said recently that application development without conventional programming "is what drives IBM's data base philosophy." What are IBM's intentions in the data base area?

I think IBM wants to substantially improve its present product line. At some point in time in the future they want to introduce relational capabilities but at the same time to support their DL/1 customers so that programs built with DL/1 data bases continue to work.

IBM wants the data dictionary built into the data base facilities. They want eventually to get a higher level of hardware involvement, in other words to substantially improve the hardware for data base usage, including expanding what they've already done with the System/38 where the entire DBMS and the dictionary are built into the hardware and the microcode.

The most important part of IBM's strategy relating to data bases is to get high-level languages which will create applications in a short time with a very high level of productivity on the part of the application developer. These will be both languages for the professional and also languages for the end user.

Today they have four such languages: CIS, Query by Example, ADF and DMS. I would expect that the future will see substantial upgrading of all of those four and possibly the introduction of new ones. Certainly one can see a lot of improvements which are needed in today's ADF, but at the same time one can find examples of ADF being used with an *extremely* high productivity of application development compared with the development that would be possible in languages like Cobol.

Therefore if we look at data base systems five years in the future, I'm not sure that people will still be arguing about the internal structure of the data base. What they'll be arguing about is how the data base is perceived by the end user or the professional who wants to develop applications. That's really saying that the end user or the application developer languages become the tail which wags the whole data base dog. It becomes the thing which drives the architecture and possibly ultimately drives the machine design.

"Data base in the 1970s has not been 100% successful," you said recently. What has been its chief problem and what makes the critical difference between a successful and unsuccessful implementation?

There is not a simple answer to that because there have been several different problems.

But I think perhaps the most important is the question of getting logical data base structures which really contain the information you're going to want for present and future applications.

So the data modeling becomes absolutely critical. I use the phrase "stability analysis" and have a substantial number of steps which I would want to apply to a data model to achieve the objective of making it as stable as possible before it is cast into the concrete of IMS. You are never going to be able to create a data base which will never change, but there are certain types of changes which are going to force you to rewrite your old application programs and certain types of changes which won't. You want to minimize the changes that will force you to rewrite your old application programs.

Now, if you build a data base system which is not stable or does not meet the needs of future applications, then you suddenly find yourself faced with an agonizing decision. You want to use the data which is in the data base, but you want to use it in a new form. The present data base structure doesn't allow you to do what you want with it. So what can you do? You must either

Now is the time to face the . . .

CHALLENGE OF APPLICATIONS

SNA — Taught by: Frank & Guests

Learn SNA

SNA is the blueprint for IBM's telecommunication's future. It's essential to understand it today. Prepare for it. Evaluate it. This seminar provides a complete overview of SNA. Learn user considerations, alternative network strategies and design concepts. Dr. Howard Frank, President of Network Analysis Corporation, and the guest speakers are all leading network analysts. Send today for full seminar information.

October 6–8, 1980
NEW YORK

Taught by: Wayne Lea

Learn Machines that Understand Speech

Speech recognizers are already in use today, but where will this lead? Experts agree that speech recognition is not a DP parlor trick, but the wave of the future. Learn the techniques of speech recognition. Learn where it is leading. Prepare for it now! Dr. Lea has published extensively about various speech recognition topics during the past 13 years. Send today for full seminar information.

October 22–24, 1980
LOS ANGELES

Taught by: Lucky & Green

Learn Data Communication Services and Protocols

So much is happening! So many computer manufacturers. Different common carriers. A confusing proliferation of networks and protocols. This seminar is designed for analysts who want to examine and understand the many alternatives available. Learn the options. Learn the benefits. Learn the pitfalls. Robert Lucky and Paul Green are two of the industry's leading experts in this discipline. Send today for full seminar information.

October 20–22, 1980
SAN FRANCISCO

Advanced applications seminars taught by leading experts from

TRANSFER TECHNOLOGY INSTITUTE

Taught by: Norman Abramson

Learn Satellite Data Networks

Astounding reductions in satellite system costs have taken place. Now learn how your DP needs can take full advantage of the technological advances of satellite data networks. Learn about regional and domestic systems. Learn about small earth terminals, costs, prices, tariffs, technical planning and timing. Dr. Norman Abramson, innovator of the ALOHA System, is a world authority on satellite systems. Send today for full seminar information.

October 13–15, 1980
WASHINGTON, D.C.

Taught by: James Emery

Learn Strategies for Distributed Computing

The potential exists for tremendous efficiency and savings with distributed systems, but only if you understand the planning techniques. What are the trade-offs with distributed systems? Is there unnecessary duplication? How can you plan for the best distributed system? Professor Emery, a leading authority on management information systems, will give you a comprehensive view. Send today for full seminar information.

November 12–14, 1980
NEW YORK

Taught by: Robert Holland

Learn Managing the Data Base Environment

The most elegant data base can fall apart when it meets the increasingly complex demands of corporate information needs. Learn how to create the data structure that is *productive* and responsive. This seminar for managers shows the planning techniques that help you save time and labor in the creation of *practical* data bases. Dr. Holland is a world renown data base expert. Send today for full seminar information.

December 8–10, 1980
WASHINGTON, D.C.

SEMINARS OF EXCELLENCE

CALL TODAY:
(213) 394-8305
Ask about these and our other quality seminars.

Please send me detailed infomation about these
APPLICATIONS SEMINARS

CW 9/17

Name
Company
Address
City State Zip
Phone

TECHNOLOGY TRANSFER INSTITUTE
P.O. Box 49765
Los Angeles, CA 90049 (213) 394-8305

"The future of data base machines is very bright. It's the way to go. But it won't all happen next week."

restructure the data base, or else you create a new data base. If you restructure the data base, then you have to rewrite the application programs which are built on top of it.

Corporations look at those alternatives and say they can't possibly afford to rewrite all of the applications programs, so they create another data base . . . and then another data base . . . then another data base . . . then another data base. By the time they've finished they've ended up with something that is like a file system rather than a data base system. And it's got all the maintenance problems of a file system.

"The proper place for DBMS is not in software but in hardware," you have said. Is software on a chip viable? What are its constraints?

You've already got one or two machines with DBMS in hardware. They are the System/38 and Microdata's Reality.

You can greatly improve the efficiency of machine usage by building a machine which can execute data base operations. That argument becomes still stronger as you move to the powerful query languages where you'd like to have the relational data base with the capability to automatically do relational algebra operations or where you have a language which triggers many secondary key operations in which you are doing much secondary index searching. You really need hardware assistance to make that fast and efficient.

The future of data base machines is very bright. It's the way to go. But it won't all happen next week.

What effect will the rapidly changing hardware technology have on data base design and implementation?

The very important thing in data base design is to get the logical structure of the data base right, getting the data models and getting the corporate-wide planning of the data base facilities put together.

A lot of people ask what they can do to protect themselves from changing technology in data base. I would say the best thing is to get a good methodology for corporate strategic planning of the information resources you need and a methodology which translates that into the most stable, logical data base model or set of models. If you do that well, then the structure will still be valid when the hardware changes fundamentally. So it will be a relatively easy job to translate what you're implementing today into the hardware of the future and the software of the future.

I founded Database Design, Inc. in Ann Arbor, Michigan, with Dixon Doll and Bob Holland in order to help users solve these problems. The field desperately needs hard and sound methodologies for data base planning and design.

You believe in the bright future of bubble memories, calling them "one of the most important memory technologies in the future." But bubbles seem to have been accepted rather slowly in the marketplace. With the falling price of RAMs, can bubbles remain in the memory race?

Certainly. We still don't have a mass production factory. With RAMs we are very high up the learning curve, whereas with bubble memory we're very low on the learning curve.

So I would expect that bubble memory technology is going to drop in cost much faster than silicon RAMs over the next 10 years. Bubble memories also have the advantage that they are nonvolatile.

The Japanese are starting to export their own technology as they overhaul their industrial structure. They are becoming increasingly competitive in computer, telecommunications and microelectronics. Some observers say

Be the first to . . .

LEARN THE SECRETS OF DP PRODUCTIVITY

Discover these important, moment-of-the-art seminars from TRANSFER TECHNOLOGY INSTITUTE

Taught by: Joe Fox

Learn Software Principles for Management

Do you need to understand how software works? Why it costs so much? This compact and productive seminar introduces managers to the principles, vocabulary and function of software. At IBM, Joe Fox managed over 4000 people in such projects as Apollo, Skylab, and the Space Shuttle. Send today for full seminar information.

Oct. 27–29 — NEW YORK
Dec. 3–5 — LOS ANGELES

Taught by: Jewett Shrago

Learn Integrated Telecommunications Management

So many options. So many benefits! So much to identify. Learn the fast, efficient techniques for telecommunications management. James Jewett is an international authority in telecommunications. Jacqueline B. Shrago is well known for her work on the optimization and analysis of voice networks. Send today for full seminar information.

Oct. 2 — LOS ANGELES
Nov. 6 — BOSTON
Dec. 5 — HOUSTON

Taught by: Couger Zawacki

Learn Motivating Computer Personnel

Great equipment but productivity lags? Motivation of skilled DP personnel can be the secret to success, efficiency, on-time performance and rock-bottom failure rates. Achieve maximum output from lowest possible labor costs. J. Daniel Couger and Robert A. Zawacki have both done extensive research on the motivational factors and norms of DP personnel. Send today for full seminar information.

Oct. 15–17 — WASH., D.C.
Nov. 12–14 — SAN DIEGO

Taught by: Tom Gilb

Learn Design by Objectives

What happens when you mistakenly design by function alone? How do you identify and *control* your system? Use of the Management by Objectives approach to DP planning brings productivity, clarity, efficiency and success. Formerly with IBM and author of over 100 publications, Tom Gilb leads you through the procedures that apply Design by Objectives to your DP challenges. Send today for full seminar information.

Oct. 22–24 — NEW YORK

Taught by: Farber Uhlig Bair

Learn the Automated Office

We now know that *how* we manage an office changes with new technology. Already, computer-based tools are in the hands of individual works. What efficiencies can we expect? David Farber, Ronald Uhlig, and James Bair created this seminar for managers who want to learn the way office systems can, could, and should be built. Send today for full seminar information.

Dec. 1–3 — SAN FRANCISCO

Taught by: Randall Jensen

Learn Software Engineering

Take your choice. Approach software like blind men examining an elephant, or learn the systematic, efficient ways of identifying, defining and creating *exactly* the software system you need. Dr. Jensen is an international authority and lecturer in the field of software. He leads you through the development of the *system* that makes software creation simple, predictable and efficient. Send today for full seminar information.

Oct. 13–15 — LOS ANGELES

SEMINARS OF EXCELLENCE

CALL TODAY:
(213) 394-8305
Ask about these and our other quality seminars.

Please send me detailed infomation about these
PRODUCTIVITY SEMINARS

CW 9/17

Name
Company
Address
City State Zip
Phone

TECHNOLOGY TRANSFER INSTITUTE
P.O. Box 49765
Los Angeles, CA 90049 (213) 394-8305

WANG USERS

AIMS is a "user friendly" application generator and DBMS.

- Slashes development time and costs
- Excellent for prototyping
- Non programmer dependent
- Over 150 installations
- Videotape demo available

AIMS PLUS

SOLUTION SOFTWARE

P.O. BOX 17247 • AUSTIN, TEXAS 78760 • (512) 385-6782

SYSTEMS SOFTWARE PROFESSIONALS

Gary Nelson & Associates specialize in recruiting software and hardware Managers and Engineers. Many of our client companies are advancing the state-of-the-art in areas such as systems architecture, distributed data processing, languages and office automation. Included are some of the top growth companies of the 1970's and a number of exciting new ventures for the 1980's.

Current San Francisco Peninsula openings include:

MANAGEMENT to $70K

- Director of Engineering
- Manager of Design Engineering
- Manager of Software Development
- Manager of File Systems

ARCHITECTURE, DESIGN & DEVELOPMENT to $50K

Operating Systems • Languages & Compliers
Database Systems • Communication Software
Microcode & Diagnostics • Office Automation

GNA

We invite you to send your resume or call Mr. Gil Siegel in strict confidence, 10050 N. Wolfe Road Suite 275 Cupertino, CA 95014 (408) 255-7400.

Gary Nelson & Associates, Inc.

OFFICES: Portland/San Francisco/San Diego

To each his own.

GET OFF THE ROUTING LIST AND ON TO THE SUBSCRIBER LIST.

CIRCULATION DEPT.

COMPUTERWORLD

THE NEWSWEEKLY FOR THE COMPUTER COMMUNITY

375 Cochituate Road, Framingham, MA 01701

(617) 879-0700

THE RING OF JUPITER AND THE VOLCANOES OF IO

Discoveries that have created excitement around the world. You can be part of the excitement in the future of JPL. Just forward your resume.

Right now we are looking for:

MEMBER OF TECHNICAL STAFF

Advanced degree in computer science/engineering or equivalent. Four yrs. minimum experience with IBM 370, DEC PDP-11 applications, and software development. A background of computer systems hardware and signal processing desirable. Responsible for integration of an array processor system to provide improvements in certain classes of image processing.

Call Paul Jepsen at:
(213) 354-6012 collect
for technical details or
send resume to Dept. D11

JET PROPULSION LABORATORY
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91103

THERE'S ONLY ONE JPL

An Equal Opportunity Employer M/F

Make The Right Move

Check Dataware for an economical answer to your tough conversion problem. Here are some winning moves ...

(any) COBOL to (any) COBOL

One of the many successful Translators offered by Dataware is our COBOL-Converter, a table-driven conversion system designed to convert COBOL programs from one version or level of COBOL to another.

RPG to COBOL
or for those PL/1 Shops
RPG to PL/1

Our Translators can handle most any level of RPG (SYS/3, Mod 20, etc.) and achieve an extremely high percentage of automatic conversion (approaching 100%) of the source code.

SPS/1400/1410 AUTOCODER to COBOL

The industry's most successful translator will eliminate the burdens of emulation and costly maintenance of second generation language.

For more information please call or write today!

The Conversion Software People

Dataware, Inc.

2565 Elmwood Ave., Buffalo, NY 14217 (716) 876-8722 / TELEX: 91519

transcomm

BUSINESS APPLICATIONS

TOLAS – II

DEC RSTS/E

ICP Million Dollar Award Winning Products, from the World's *First* Commercial RSTS Service Center.

DISTRIBUTION SYSTEMS

- Order Entry/Invoicing
- Inventory Management
- Sales Analysis
- Purchasing

FINANCIAL SYSTEMS

- Accounts Receivable
- Accounts Payable
- Financial Reporting + FINAR
- Payroll • Fixed Assets

TRANSCOMM DATA SYSTEMS

412-963-6770 — 1380 Old Freeport Road, Pittsburgh, PA 15238

415-433-7123 — San Francisco, CA

AUTHORIZED digital COMPUTER DISTRIBUTOR

"America is in danger of losing part of its lead to Japan if it doesn't fight back aggressively with lots of money for R&D."

the world ought to brace itself for an onslaught of Japanese high-tech products. Is the U.S. in any danger of losing its technological edge to the Japanese?

I think it's in danger of losing part of it. The U.S. has absolutely led the world with chips for the whole history of chips. But now Japanese are talking about a memory chip of 256,000 bits. They haven't actually manufactured it yet, but they have described the technology in public. So it's beginning to look as though they are not just copying. They are going to be able to produce chip technology which *could* become better than Silicon Valley. And that is very dangerous. It means that Silicon Valley would no longer be providing the crude oil, if you will, of the electronics industry.

The Japanese are going to be very formidable competitors. They have gone a very long way in the use of industrial robots and are obviously putting a large amount of money into the technology. I think we ought to regard Japan as a major warning sign that America has got to continue driving forward in improving its research. It is in danger of losing part of its lead to Japan if it doesn't fight back aggressively with a lot of money for R&D. America should be as aggressive as it possibly can in attacking the Japanese threat.

A recurring theme in your writing is the potential abuse of technology by government. "Computers, like other inventions that change civilization, are a force for great good or great evil depending on how countries use them," you wrote in one of your books. How are governments using computers these days — for good or evil?

I think it differs very much from one government to another. The governments on the other side of the Iron Curtain are using them in quite different ways from the governments in Western societies. I don't think there is much to worry about in the way the Western governments are using computers except that they have permitted the bureaucracies to grow to much larger sizes than they would have done otherwise.

What about the invasion of privacy?

One finds occassional horror stories. There is one airline which is a government airline, as many airlines are outside the United States. Somebody sat me down at a terminal of that airline and explained to me that if, as a passenger of that airline, you are exceptionally insulting to the air hostesses, they put you on a blacklist. And that blacklist is then used so that when you try to make a booking on the airline again, the computer refuses you a seat.

At the terminal I was shown how to display that blacklist. The amusing thing was that many of the people on the blacklist were people whose names you would know! I wonder if that would happen in a profit-making airline.

We talked earlier about distributed data processing. Does DDP make a corporation easier to manage?

I think it will make it easier to manage if it's done well. Because DDP can follow the corporate structure.

Most computing has forced the flow of information into a structure which is quite different from the natural organization of the corporation in question. Whereas distributed processing enables you to put computers and information sources in the departments where they are used.

But we are not yet at that utopia, are we?

Not yet, but machines are dropping in cost.

Incidentally, in moving to that utopia I think the last thing you want to do is to have complete freedom of letting anybody put any data they want together in a computer system. You do need to understand that a corporation

Take everything you knew about DP Productivity and FORGET IT!

Attend The James Martin Seminar with its new emphasis on DP Productivity
SEND FOR FULL INFORMATION

JAMES MARTIN SEMINAR

Management Strategy ■ Distributed Processing ■ Data Base ■ Office-of-the-Future ■ Computer Networks

Productivity is the real magic to your DP. Does it produce? In five intensive days of lectures by James Martin you have the facts, techniques and methods that evaluate, design and modify your DP system.

APPLICATION DEVELOPMENT WITHOUT PROGRAMMING IS THE KEY TO PRODUCTIVITY.

The James Martin Seminar focuses on this technique for productivity. Learn how the DP revolution gives you tremendous new opportunities for corporate DP strategies. Achieve gains. Evaluate changes in traditional DP methods. *Step out of the trenches for five days.* Share James Martin's new perspective.

HOUSTON	October 20–24, 1980
WASHINGTON, D.C.	October 27–31, 1980
HONOLULU	November 10–14, 1980
BOSTON	November 17–21, 1980

ONLY JAMES MARTIN COULD GIVE YOU THIS SEMINAR.

James Martin is the world's best selling author of computer books. His reputation as one of the finest lecturers in the DP field is unchallenged. *Computer Weekly* described James Martin as "the computer industry's most widely read author and best attended lecturer."

James Martin's seminars always include his opinions of recent developments and predictions for what is to come in the DP field. In the past, his predictions about technology have proved uncannily accurate.

NOTICE

If you have attended the James Martin Seminar in the past, then send for information on this new, updated presentation. Learn how much more has happened since you last attended!

Call or write today: (213) 394-8305

CW 9/17

SEND FOR FULL INFORMATION ABOUT THE JAMES MARTIN SEMINAR.

Technology Transfer Institute
P.O. Box 49765
Los Angeles, CA 90049 (213) 394-8305

TECHNOLOGY TRANSFER INSTITUTE
P.O. BOX 49765, LOS ANGELES, CA 90049 (213) 394-8305

Name
Company
Address
City State Zip
Phone

"End users will have a lot of compatibility problems in the '80s. They're getting worse rather than better."

is one entity and it has one balance sheet; therefore you need to make sure that the different computer systems do fit together to contribute to the overall corporate good. That needs corporate-wide information engineering.

Have we reached or even neared the natural physical limits of our technology?

Oh no, nowhere near, nowhere near. The theoretical information transmission capability of an optical fiber is thousands of times higher than what we are using today. But we couldn't possibly build circuits which are good enough to use that transmission rate. We can make an optical fiber about as thin as a human hair and so into one cable one could get many hundreds of thousands of optical fibers. In terms of telecommunications, the capability for wiring society with very advanced channels is so *enormously* greater than what we've done today that it takes the richest imagination to conceive what that would do to society and how society would use it.

If you think of the size of the brain of the micro-ant and how much logic must be in that ant, then compare it with today's chips — that gives you an indication of just how enormously far away we are from nature. Compare the protocols of SNA with the protocols of micro-ants!

Communications network architectures can provide impressive capabilities. But their drawbacks include high conversion costs, "lock-in" to the products of a single manufacturer and incompatibility with networks based on other architectures. Is there a way out of the maze for end users?

I think end users and corporations in general are going to have a lot of compatibility problems in the 1980s. Compatibility problems are getting worse rather than better.

In the long run there is a way out of the maze, and that is standardization. We've already got some good standards at lower levels, like HDLC and X.25. It's clear that X.25 needs improving, and it's also clear that we need standards at a much higher level than X.25. But as one gets those standards in an age of vastly improved chips, the standards will become converted into chips.

One of the phrases in the industry now is PHD, which (believe it or not) stands for Protocol-Handling Device. That is a microelectronics device which can handle standard protocols like X.25 or HDLC. Protocol conversion machines and protocol handling devices are the long-range solution.

IBM's SNA is probably one of the most well-known and least understood architectures. How does SNA differ from other architectures, such as DCA and Decnet?

Sometimes the differences are rather subtle. SNA is an architecture which needs a host computer to establish a session, whereas Decnet is an architecture in which you can go from any machine to any machine.

The transmission subsystem of SNA is different from the internationl standards based on X.25. The highest layers — the layers which IBM calls "function management" — have a lot of functionality and a lot of capability relating to presentation services. These enable one to employ intelligent terminals and intelligent devices to convert the data stream, to have virtual terminals, to have logical display space, to do editing, to compress the data and to encipher the data. Some other architectures don't have as much capability at the higher level layers.

On the other hand, DEC has strongly emphasized the use of remote files. So with Decnet you can have a Cobol program which says GET a record when that record is in a file 1,000 miles away.

IBM's SNA is more complex than many of the network architectures. Part of the reason for that is that it does have more functions than some of the others. But if it were reengineered

Model 204 DBMS.
CCA's Revolutionary Idea.

We think a DBMS should have excellent performance and be easy to use.

Not a revolutionary idea? Guess again.

Only Model 204 can do it. There are some inverted-file systems that are easy to use. But their performance falls apart when the database gets large or the number of terminals begins to grow.

And then there's IMS. It performs pretty well, if you keep it tuned up. But it's hard to use. Which means it takes large teams of programmers to develop and maintain applications.

But with Model 204 you get excellent performance, even with large databases and many on-line terminals. And using Model 204 is exceptionally easy. The leading software rating service gives Model 204 a perfect rating—4 out of 4—for ease of use.

So with Model 204 you get the best of both worlds. Excellent performance and ease of use. And that's why Model 204 is being used extensively by some of America's largest companies—for financial applications, inventory control, correspondence tracking, personnel data, project management—in fact, every kind of corporate database. It will pay you to check it out.

Model 204. Revolutionary!

For details, clip your business card to this ad and send to:
Computer Corporation of America
Database Products Division
675 Massachusetts Avenue
Cambridge, Massachusetts 02139
617-491-7400

Model 204

Computer Corporation of America

Cambridge, New York, Houston, San Francisco, Chicago, Washington / International affiliates: Tokyo, Berne, Milan, Toronto

"SNA has an enormous future. And I think we'll see very major changes in SNA as we progress through the 1980s."

today, it could be simplified.

Given the excitement that X.25 seems to have sparked among users, does SNA have a future?

Oh yes, SNA has an enormous future. One can think of SNA as a common language throughout IBM, a common set of formats and protocols which will enable one to build an almost infinite diversity of different systems from many different components.

This is very important to the future of IBM. Architectures like that are important to the future of every computer vendor. That is not to say that SNA is not going to change. And I think we will see *very* major changes in SNA as we progress through the 1980s. One wants changes which will increase the diversity of the higher level functions and one also wants changes which will make SNA conform to the international standards. I would expect that we will be seeing both of those pretty fast — in two or three years.

You are very excited about space satellites and space manufacturing. Where do satellites fit into the grand communications scenario?

If we talk about telecommunications, one important thing to say is we haven't launched any communication satellites yet which use the potential of the technology. We haven't yet seen a satellite which was designed for space shuttle launch.

What the engineering is saying to us is that there are great things you can do which you haven't yet done, and you can do them during the 1980s if the lawyers don't cripple you first.

What effect do you believe the recent MCI antitrust victory over AT&T will have on the structure of the telecommunications industry?

I imagine it will be great encouragement to MCI and the other specialized carriers in their competition with the established telephone companies. I expect much more aggressive competition.

Your predictions about technology have been on the mark for 20 years. In 1960 you predicted the growth of teleprocessing, on-line storage and real-time systems. Later you predicted the microcomputer boom, communications satellites, optical fibers and personal computers. What's next?

We've mentioned pocket terminals. We've mentioned viewdata-like terminals. The growth of business graphics is important. Data base machines and languages. Ubiquitous public data networks.

We've talked about the potential of communication satellites.

We've talked about industrial robots. This will be a *major* growth industry throughout the next 20 years. And because industrial robots become economic, product designers must redesign their entire products to take advantage of totally different manufacturing techniques.

We've talked about the growth of chips. We're going to see much more dense memory and logic chips. We're certainly going to see 10M-byte chips, though that may take 10 years or more. That means the cost of main memory is going to *plunge* down.

The military is asking for an IBM 370/158 processor on a single chip. I expect they will succeed in achieving that in the second half of the 1980s. The impact that that will have on the possible range of products that you can put on the market at low cost will be just gigantic.

The videodisk. It is slowly becoming reality. Eventually it is going to become a mass consumer product. Philips has videodisks in their laboratories which are designed to store data and to be an important component of the office of the future. The price per bit on those videodisks can be less than

ORACLE

THE RELATIONAL DATA BASE MANAGEMENT SYSTEM

INTRODUCING A NEW GENERATION OF DBMS TECHNOLOGY

ORACLE is the only commercially proven Relational DBMS. First installed in June 1979, ORACLE is now in daily use at over 35 customer sites.

ORACLE's user interface is SQL, the Relational Data Language developed by IBM Advanced Research. All SQL Query, Data Manipulation and Definition facilities are available from Online Terminals as well as from programming languages such as COBOL, FORTRAN, etc.

ORACLE is a High-Performance System consisting of 100% Reentrant Code, allowing multiple online users to concurrently update or query the data base.

ORACLE FEATURES:

- An **Integrated Data Dictionary** that centrally defines all information stored in the data base.
- An **Interactive Application Facility** that allows the end-user to generate online data entry and data maintenance applications without the need for conventional programming.
- A **Report Writer/Word Processor** that allows formatted data from the data base to be interleaved with formatted text within a single report.

ORACLE is available today on DEC VAX—11 & PDP—11 CPU's; IBM Mainframes.— March 1981.

See Oracle demo at Software Info & Federal Computer Conference

RSI

Relational Software Incorporated 3000 Sand Hill Road, Menlo Park, California 94025 415-854-7350

"I believe teachers will make more money than film stars because of videodisk and telecommunications technologies."

1/10,000th of the price per bit on the cheapest high-capacity magnetic disk systems of today. Not overnight, but by the second half of the 1980s the videodisk is going to become a major technology.

That will give rise to whole new industries. You suddenly for the first time make interactive television real.

I believe it has *huge* social potential in education because it gives you the capability to have education with Hollywood-size budgets in which the machine stops periodically and asks the student questions. He responds to those and then you switch videodisk track depending upon the response. It also enables the student to stop the program at any time and recap — to study the diagram he was shown two minutes ago and to make notes on it before continuing the program. When you can do that, then interactive television training becomes much more potent than it is today.

I believe that the 1980s will probably be a decade in which more Americans become new millionaires than any other decade in history — even if we adjust that remark for inflation. The reason is that the technologies I've just described open up enormous business opportunities. Many of those, unlike the past, are business opportunities which can be grasped by the small entrepreneur. The programmer, if he really thinks about the marketplace, has an opportunity now to create programming products which will have gigantic sales, and he can do it without having a big corporation. The person who is designing chips has the capability to again design products — household gadgets, toys, devices for the office of the future — with enormous potentials.

One thing now happening in the microelectronics industry is that it's becoming easy to design a new product. If you're a programmer and you write in Basic, you can then easily acquire the professional help to convert your Basic program into a chip or a microcode program so that you can put that into a mass-produced, inexpensive device.

If you look at the videodisk for television programs, the amount of software that is needed for that is just gigantic. Once again the best conventional or interactive programs are going to sell in large quantities, and you don't need a large corporation to create those.

I believe the best teachers will make more money than film stars by the end of the 1980s because of the videodisk and telecommunications technologies.

One informed observer once said that there will be no major software breakthroughs before the year 2000. Do you agree?

Oh, I disagree enormously. I think there are gigantic software breakthroughs to come. If one looks at the potential of developing applications without conventional programming, if one looks at the higher level data base languages and interactive graphics — these are indications that we have the opportunity within a relatively short period of time to greatly increase the productivity of application development.

If there is any one last message I would want to stress, it is the whole question of productivity. This is becoming more and more important; as the machines drop in cost and the chips increase in power, then the cost of people becomes so out of proportion to the cost of machines. At that point it is no good thinking of improving productivity by 10%; you've got to improve it by a factor of 10.

So all DP executives ought to be addressing themselves to the question of how we get a step-function change in productivity in application development. That's the subject of the series of seminars I will give in the fall.

Where will James Martin be in 10 years?

God knows.□

The Winning Combination

Will Be In Chicago
On September 23rd to 25th
At The Software
Info Exposition

Come Visit
Stall 111
And See Some
Real Winners!

CustomORDER™...

On-line ORDER ENTRY, INVOICING and INVENTORY CONTROL in the saddle, making split-second decisions... getting orders accepted and moved along to the finish line... checking automatically for any obstacles or hindrances communicated by CustomAR...updating and positioning the available resources instantaneously... and then breaking away from the competition to take the lead. Nice job.

CustomAR®...

The thoroughbred on-line ACCOUNTS RECEIVABLE AND CREDIT MANAGEMENT...receiving input... executing commands automatically... assuming complete control over the field... changing within seconds to new situations... capable of transferring positions with ease... transmitting messages for instant decision making... always ready for queries by CustomORDER.

Together they're the winning combination, finishing first to improve the cash flow and profitability of the investors. Nice job indeed! You too can be in the winner's circle. Call or write today for additional information on either or both of these champions.

NEWS FLASH!!

CustomCONTROL™... our On-line **MiniComputer** MANUFACTURING AND DISTRIBUTION SYSTEM with a proven track record will be presented for the first time in the Chicago area!

computeristics® inc.

Write or call Chet Habegger
2 Skiff St., Hamden, CT 06514
(203) 288-4885

COMMUNICATION NETWORKS

Conference & Exposition

January 13-15, 1981
Albert Thomas Convention Center
Houston, Texas

3rd Annual ComNet Draws Top Professionals...

Just three years old, but the show and conference have already won an international audience that's concerned with TOTAL communications. . .the unified voice and data NETWORK that digital technology will put in place. The top corporate telecommunications executives are drawn to COMMUNICATION NETWORKS because they know the program will address ALL of their technical and managerial needs.

Over 70 Conference Sessions and an Exhibit Hall packed with the industry's elite manufacturers

Leaders in telecommunications management, technology and applications make up the COMMUNICATION NETWORKS speaker list. You'll meet them at more than seventy planned sessions.

Over five dozen leading U.S. and international experts in network technology and management are chairing and putting together informative sessions offering the practicing telecommunicaitons professional the best program in the industry.

Call (800) 225-4260 for latest program information

Here's the Tip of the ComNet Iceberg

NETWORK ARCHITECTURE
NETWORK POLICY AND MANAGEMENT FOR U.S. AND INTERNATIONAL USERS AND CARRIERS
NETWORK OPTIMIZATION AND OPERATIONS
NETWORK SOFTWARE
NETWORK HARDWARE
NEW DEVELOPMENTS
SATELLITE DEVELOPMENTS
INFORMATION APPLIANCES
NETWORK APPLICATIONS
NETWORK ECONOMICS
OFFICE OF TOMORROW
SECURITY, EXCRYPTION AND PRIVACY PROBLEMS
AND MORE!

Register Now

MAIL TO: COMNET '81, 375 Cochituate Rd., Framingham, MA 01701

I'm interested in information on:
☐ Conference Registration ☐ Exhibit Space

Name _____
Title _____
Company _____
Address _____
City/State/Zip _____
Telephone _____

ADVERTISERS' INDEX

Aims Plus 116
Allen Services 7
Altergo 57
Applications Software 112
Applied Data Research ..... Cover 2
Ask Computer 110
Art Benjamin & Associates 72

BBN Software 84

Cambridge Systems Group 64
Capex 35
Cincom 62, 63
Columbia Computing Service 28
Computer Associates 43
Computer Concepts 79
Computer Corp. of America 118
Computeristics 120
Cullinane 102
CW Conference Management Group 121
Cyborg 108

Data Processing Design 106
Data Structures 86
Data Systems for Industry 86
Dataware 116
Digital Equipment Corp. 26, 27
Digital Research 83
Dylakor 81

Eastern Airlines 93
EDP Security 28
Electronic Data Systems 58
Execucom Systems 70

Financial Software 24
Finar Systems 79

Goal Systems 25
GTE Automatic Labs 68

Henco 81

Improved System Technology 82
Infodata 14
Information Sciences 100
Innovation Data Processing 29
Intel 90, 91
Interactive Management Services 98
International Computer Program 77
International Database Systems 6
International Systems 28
ISSCO 61

Jet Propulsion Labs 116
Johnson Systems 3

Lloyd Bush & Associates 22

Maestro Systems 94
Mathematica 2
MCBA 76
McCormack & Dodge ..... Cover 4
Memorex/QMC Group 54
MicroPro 74
Microsoft 101
Modems Plus 88
Morino 107
B.I. Moyle & Associates 98
MSA 95

National Systems & Services 78
NCA 34
NCR Corp. 71
Gary Nelson 116
Nixdorf Computer Software 103

On-Line Business Systems 73
On Line Software International 113

Pace Applied Technology 97
Perkin Elmer 20, 21
Professional Computer Resources 99

QED 42, 84

Relational Software 119
Ross Systems 75

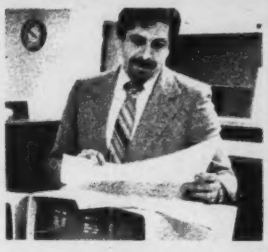
SAS Institute 80
Sercon 60
Small Business Applications 39
Software AG ..... 122, Cover 3
Software House 66, 67
Software International 4
Software Module Marketing 12
Source EDP 96
Superior Software 28
Synergetics Corp. 40
Systematics International 111
Systar Corp. 56
Systems Research 92

Tandem Computer 104
TCS Software 105
Technology Transfer 114, 115, 117
Telos 78
Terminal Systems 28
Tesseract 28
Three M 89
Tomark 8
Tominy 13
Tower Systems 38
Transcomm Data 116
TRW 33
Turnkey 87

United Computer Systems 69
University Computer Corp. 44, 47

At INCO, we demand adaptability and ease-of-use in our DBMS

Michael Berman, Corporate Data Base Administrator



INCO LIMITED is a diversified company engaged principally in primary metals, formed metal products and electrical energy products.

Inco's DBA tells why ADABAS, Software AG's data base management system, was chosen for the expanding needs of this international firm.

"Five years ago, we chose ADABAS from Software AG as our data base management system. We demanded a highly flexible, end-user oriented DBMS that would facilitate the development and operation of our business systems. We did not have the time to design our own DBMS, or the luxury of selecting a system that would require a large staff to implement and support. We needed an easy-to-use package with the design phase already accomplished. The available studies indicated to us that ADABAS would be the best choice for our needs. Five years of experience showed us that the studies were right – and so were we in choosing ADABAS."

ADABAS delivers!

Exceptional flexibility, easy maintenance

"During the development stages of our many application systems, the specifications often changed. We were constantly adding fields and files, or changing their locations in the data base. With ADABAS, these changes are very simple to make. There are no cumbersome chain pointers, so relationships can change according to need.

"In fact, we are continuing to modify our systems to meet our expanding corporate requirements. And with ADABAS, we can make these necessary changes with little or no impact on the hundreds of batch and online programs supported by the DBMS."

Cost-effective new program development

"In planning new applications, we do not have to include the high cost overhead for restart/recovery, security, or data base loading. These utilities are all included in ADABAS already. Applications programming is faster and easier, too – because ADABAS provides the tools. This means we can concentrate on the application problem – not the data base.

"ADABAS gave us a powerful end-user query language – ADASCRIPT+, an interactive report generator – ADACOM, and an online Data Dictionary all in the same package. And recently we have added NATURAL, Software AG's instant applications language, which we expect will decrease development time. Furthermore, the ADABAS system can be fully installed in a single day!"

High company-wide productivity

"ADABAS is up and running at Inco 18 to 20 hours every working day. The system is used daily by literally hundreds of end users and programmers alike. Each day, our New York installation processes more than 400,000 data base commands from departments like accounting, personnel, marketing, customer information, shipping and billing. With all this intense activity, ADABAS is supported by only three people: a data base administrator, a data base analyst, and a data dictionary analyst."

ADABAS: Datapro Honor Roll winner for three years

At Inco, ADABAS has cut both program development time and cost. In fact, the exceptional ease-of-use of ADABAS, combined with its outstanding throughput efficiency and adaptability, helped give this product such a high user rating that ADABAS has been nominated to the Datapro Honor Roll for three consecutive years.

If you'd like to know more about how ADABAS met Inco's demand for a truly adaptable DBMS, send for Software AG's ONLINE report today. You may find that ADABAS is the ideal data base management system for a demanding company like yours, too.

SOFTWARE AG
OF NORTH AMERICA, INC

Reston International Center
11800 Sunrise Valley Drive, Reston, VA 22091
(703) 860-5050

Please send me more information on ADABAS, including ONLINE, Software AG's special user report on INCO's success with this unique DBMS.

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Telephone _____

Type of Computer _____

Operating System _____

CWA

Financial software so advanced that Encyclopaedia Britannica doesn't even want to talk about it.

Nor do any of the 1,000 companies that benefit from McCormack & Dodge financial software.

When you're ahead of the pack, the less said the better.

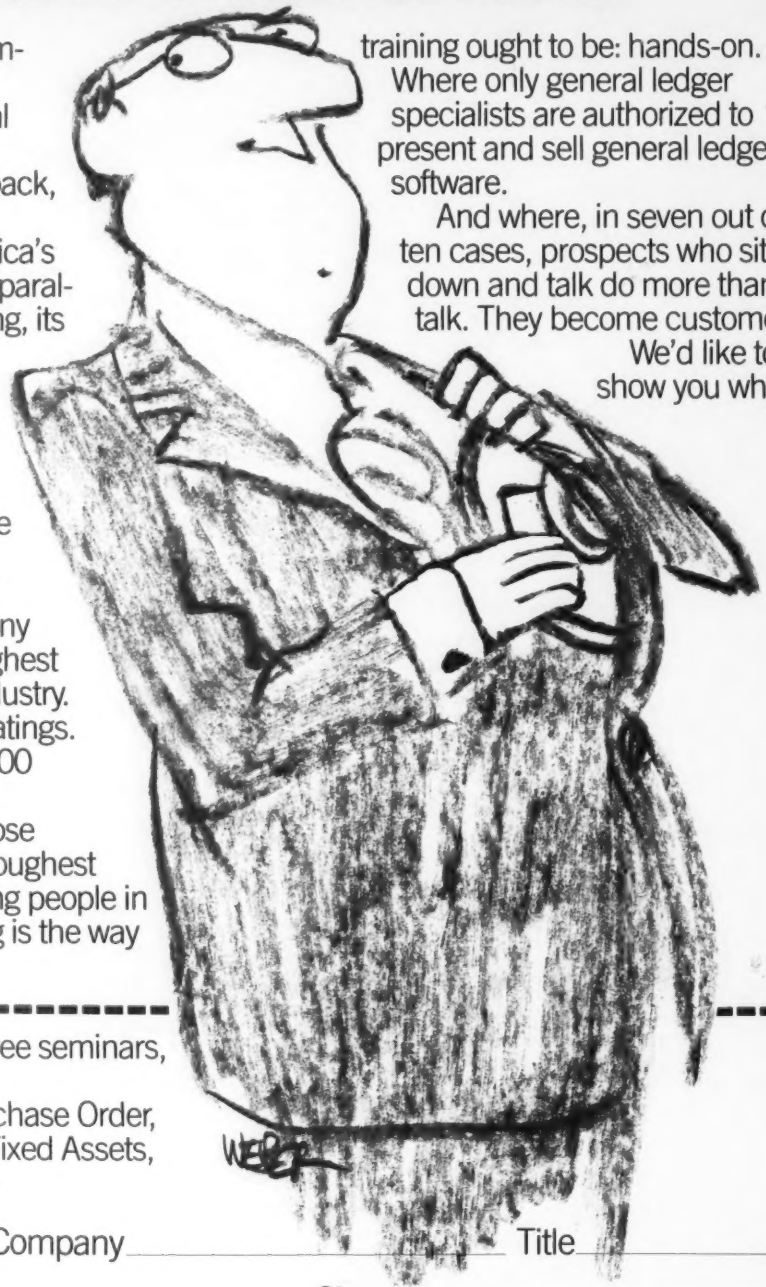
Suffice it to say that Britannica's General Ledger package is unparalleled in its fast, flexible reporting, its efficient journal processing, its comprehensive editing and extended capacity for budget control.

Behind the package is a cadre of sophisticated software professionals that meets Britannica's own standards of excellence. A software company with proven success in the highest echelons of the publishing industry. Consistently tops in Datapro ratings. Serving 20% of the *Fortune* 500 companies.

McCormack & Dodge. Whose customers are the smartest, toughest controllers and data processing people in the world. Where user training is the way training ought to be: hands-on. Where only general ledger specialists are authorized to present and sell general ledger software.

And where, in seven out of ten cases, prospects who sit down and talk do more than just talk. They become customers.

We'd like to show you why.



Please send schedule of free seminars, plus information on:
☐ General Ledger, ☐ Purchase Order,
☐ Accounts Payable, ☐ Fixed Assets,
☐ Capital Project Analysis

Name_____ Company_____ Title_____

Address_____ City_____

State_____ Zip_____ Phone_____ Computer Model_____

McCormack & Dodge

The best financial software. The best financial people.

560 Hillside Avenue, Needham Heights, MA 02194, (617) 449-4012
Atlanta (404) 997-2797, Chicago (312) 843-4300, Dallas (214) 369-7014, Los Angeles (213) 645-6382, Washington, D.C. (703) 734-0710, Toronto (416) 445-2971, New York/New Jersey (201) 528-6331. United Kingdom/Netherlands • Italy • Australia/New Zealand • Mexico • South Africa • Sweden